Connect:Enterprise UNIX®

Configuration Files Reference Guide

Version 2.4

Sterling Commerce
An IBM Company

Connect:Enterprise UNIX Configuration Files Reference Guide
Version 2.4

First Edition

Sterling Commerce, Inc.

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

Contents

Chapter 1

System Configuration Files

One of the most important tasks the system administrator has is to periodically update Connect:Enterprise. The system must be reconfigured when:

- A remote site is added, modified, or deleted
- A communications resource is added, changed, or deleted
- A host operator is assigned or changed
- Parameters for a communications session are changed
- Passwords are changed
- Auto connect parameters change
- Security parameters change
- Mailbox access permissions change
- Encryption parameters change

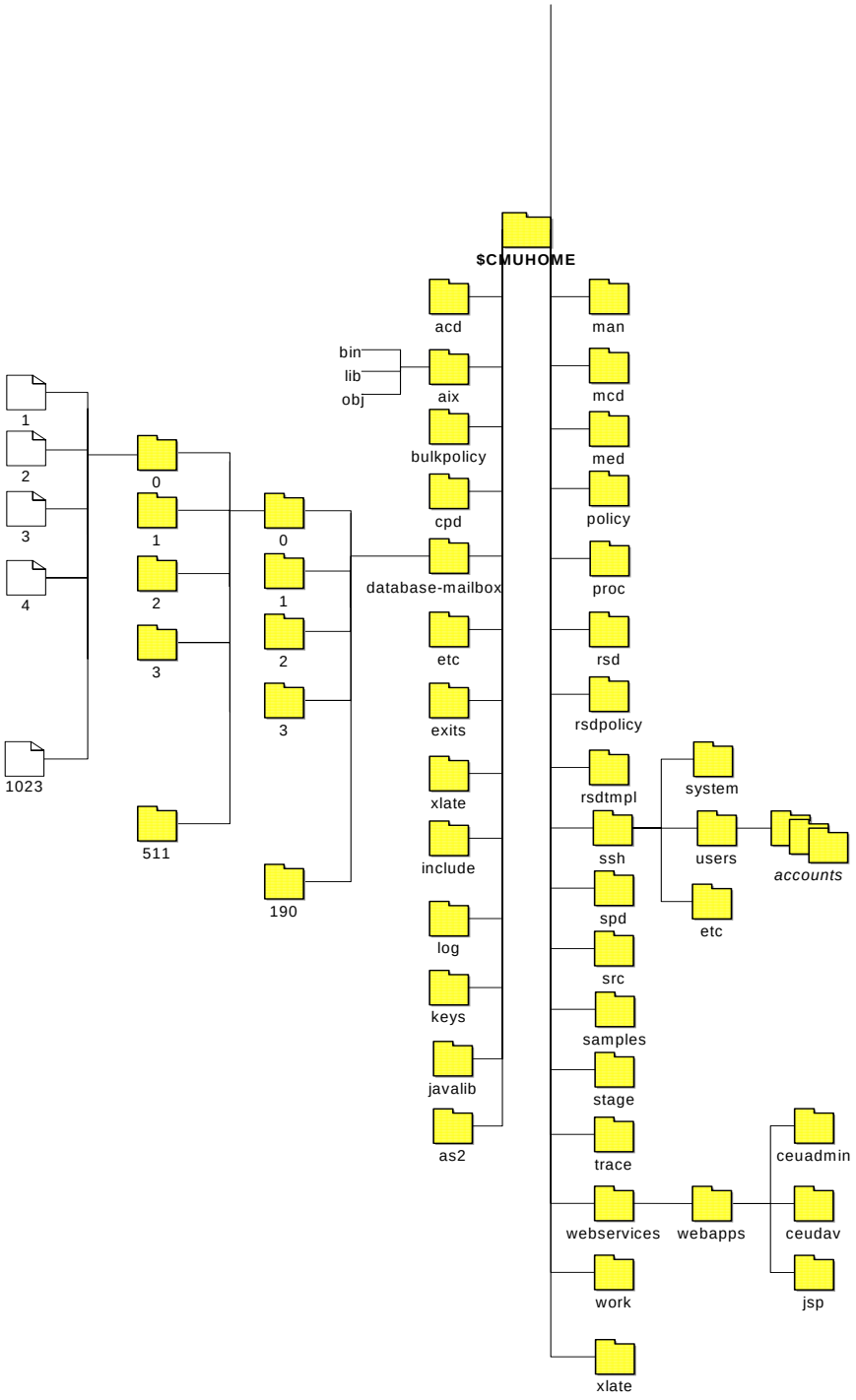
You can use the Connect:Enterprise Site Administration user interface to perform most of these tasks; however, if a configuration file contains syntax errors or elements that are not recognized, the administrator must alter the file manually. This document describes the contents of each Connect:Enterprise file that is used to configure this product.

Directory Structure

The mailbox is made up of a set of directories and a central control file that tracks the contents of the mailbox. All directories and files created by the installation process are required for system operation unless noted otherwise.

This central control file is known as the Mailbox Engine Definitions file (MED). It identifies the directories and file systems that can be used to store data. The data is stored in individual files, known as batches. The MED file points to the directory *$CMUHOME/database/mailbox* as the main data directory.

The $CMUHOME directory represents the directory where the Connect:Enterprise system is installed. The following figure illustrates the directory structure for AIX.



An overview of the contents of each directory follows.

Directory	Description
$CMUHOME/acd	The *$CMUHOME/acd* directory stores the ACD files used by the Auto Connect Daemon. Each Auto Connect list is stored as a separate ACD file. The name of the ACD file is the name of the Auto Connect definition.
$CMUHOME/aix, $CMUHOME/hpux, $CMUHOME/sun,* *$CMUHOME/linux *$CMUHOME/zlinux*	The *$CMUHOME/aix* directory is where the binary, library, and object files are kept for Connect:Enterprise. This subdirectory is named *$CMUHOME/aix* if the operating system in use is AIX. It is named *$CMUHOME/hpux* if your version of UNIX is HP-UX, *$CMUHOME/sun* if your version of UNIX is Solaris, *$CMUHOME/linux if your version of UNIX is linux, $CMUHOME/zlinux if your version of UNIX is zlinux*. It is possible to have *$CMUHOME/aix*, *$CMUHOME/hpux*, *$CMUHOME/sun*, *$CMUHOME/linux, $CMUHOME/zlinux* if you are running all operating systems on a network where NFS is used to mount the storage device to each system. Under this directory you will find the *$CMUHOME/bin*, *$CMUHOME/lib*, and *$CMUHOME/obj* subdirectories containing files that are unique for each operating system. Note: In the diagram, *$CMUHOME/aix* is represented. However, this is can be *$CMUHOME/hpux*, *$CMUHOME/sun, $CMUHOME/linux, or $CMUHOME/zlinux*.
$CMUHOME/bin	This directory stores Connect:Enterprise executables and binary files. The executables stored here depend upon what type of installation is performed.
$CMUHOME/lib	This directory contains the Connect:Enterprise API libraries.
$CMUHOME/obj	This directory contains Connect:Enterprise object files.
$CMUHOME/bulkpolicy	This directory stores the files for creating bulk policies. You can create many RSD policies from one password policy using this feature.
$CMUHOME/cpd	This directory stores the CPD files. The *$CMUHOME/cpd* directory contains one or more Bisync, Async, FTP, or SSHFTP CPD files. These files contain resource information for Bisync, Async, FTP, and SSHFTP communications, respectively. One CPD file is created for each communications daemon. The WebDAV communication port definition is defined in the Admind.xml file.
$CMUHOME/database/mailbox	This directory contains the actual batches in the repository.
$CMUHOME/etc	This directory contains sample startup shell scripts.
$CMUHOME/exits	This directory contains cmuexits.c, which is a skeleton source file users can modify, then compile and link to create user exits.
$CMUHOME/xlate	This directory contains the default ASCII/EBCDIC translation table as well as any user-defined translation tables.
$CMUHOME/include	This directory stores the Connect:Enterprise API header file. It is called cmuapi.h. The include directory also contains header files used by the exit subroutines.

Directory	Description
$CMUHOME/log	Connect:Enterprise writes all log messages to this directory. The log daemon writes to logacct.dat–data file.
$CMUHOME/keys	This directory contains the key files used for batch and password encryption. The directory exists only if you have Connect:Enterprise (Secure FTP).
$CMUHOME/javalib	This directory contains the AS2 jar files, the **cmuhttpd** and **cmuediintd** daemons, and the **as2delete** and **as2report** commands.
$CMUHOME/as2	This directory contains the AS2 configuration file. This is an XML file containing the AS2 contract, port, and proxy information. It is created at installation and is only updated using the Site Administration User Interface. Do not edit this file directly.
$CMUHOME/man	This directory contains man page help for some Connect:Enterprise commands.
$CMUHOME/mcd	The Mailbox Control Definitions directory contains an MCD file that is parsed when the Control Daemon is started. The file contains parameters that indicate what exits are to be activated and, optionally, a valid ID list of mailbox IDs that will be authorized if the **SECURITY=BATCH** parameter is specified.
$CMUHOME/med	This directory contains the Mailbox Engine Definitions file and encrypt.cfg. The Mailbox Engine Definitions file is created by cmuinit using values supplied as parameters on the cmuinit command line. These parameters define the paths to the repository directory structure and its associated control files. Encrypt.cfg defines the batch encryption parameters.
$CMUHOME/policy	This directory contains the password policy files.
$CMUHOME/proc	This directory is used to support the two AS2 daemons, EDIINT and HTTP, in a high availability environment. When ceustartup starts the EDIINT or HTTP daemons, a zero length file is placed in this directory called *daemon_name*.*pid*, where *daemon_name* is the name of the EDIINT or HTTP daemon, and *pid* is the process ID associated with the startup. High availability uses this file to test for the existence of EDIINT and HTTP daemons in case of a failover.
$CMUHOME/rsd	This directory stores the RSD files that are used by the communications daemons. Each remote site definition is stored as a separate RSD file. The name of the RSD file is the name of the remote site.
$CMUHOME/rsdpolicy	This directory stores the RSD policy files.
$CMUHOME/rsdtempl	This directory contains that templates that you can use to create RSD policy files.
$CMUHOME/ssh	This directory contains the sshd_config file which identifies trusted servers.

Directory	Description
$CMUHOME/ssh/system	This directory contains the host server key for authentication.
$CMUHOME/ssh/users/accounts	Each *account* directory contains all keys associated with that account. This includes public keys, private keys, and authorized_keys.
$CMUHOME/ssh/etc	This directory contains SSH protocol commands available that address SSH randomness.
$CMUHOME/spd	This directory stores the SPD files that are used during Secure FTP transfers. The directory only exists if you have Connect:Enterprise (Secure FTP).
$CMUHOME/samples	Contains all sample RSD files.
$CMUHOME/src	This directory contains sample API programs. Sample exit source code is also provided here as well as sample shell scripts which are executed when a specific batch of data is received.
$CMUHOME/stage	This is the default staging directory for file agent. It contains the lock file that prevents multiple file agents from running in this directory.
$CMUHOME/trace	This directory contains trace output generated when one or more of the seven daemons are started with the trace option.
$CMUHOME/webservices	This directory contains the ceuadmin.war file that must be deployed if you are using a third-party Web server to support the Connect:Enterprise Site Administration user interface. It also contains the ceudav.war file that must be deployed if you are using a third-party Web server to support the WebDAV protocol.
$CMUHOME/webservices /webapps/jsp	This directory contains all of the compiled Java Server Pages (JSP) used in the Site Administration user interface.
$CMUHOME/webservices /webapps/ceuadmin	This directory contains the directory structure used by the HTTP server. The HTTP server supports the Site Administration user interface and is an option during the installation. If you selected not to configure the HTTP server during installation, this directory does not exist.
$CMUHOME/webservices /webapps/ceudav	This directory contains the directory structure used by the WebDAV server.
$CMUHOME/work	This directory contains temporary files that are created when generating reports.
$CMUHOME/xlate	The directory contains the translate table that is referenced with the TRANSLATE parameter in one or more RSD files.

The *$CMUHOME/database* directory is part of a standard directory structure that the product uses. The *$CMUHOME/database* directory contains the internal Connect:Enterprise ISAM control files.

Under the *$CMUHOME/database* directory is a three-level subdirectory structure. This is used to optimize access to the batches.

The first level is the *$CMUHOME/database/mailbox* directory. It can contain up to 191 subdirectories numbered 0190.

The next level is within each of these 191 subdirectories, where another 512 low-level subdirectories (numbered 0–511) are defined.

The last level consists of individual files that store the data contained in a single batch. Each low-level subdirectory can contain up to 1,024 such files.

A batch data file name is its batch number. For example, batch numbers 1–1023 would be located in *./mailbox/0/0*. The maximum possible number of batches is 99,999,999 with the final batch located in *./mailbox/190/376/*99999999.

Editing Definitions Files

To edit configuration files manually, use the following instructions:

1. Open the designated definition file by using a text editor, such as vi. Each type of definition file is found in a separate directory, as described in the following table.

File	Directory
Mailbox Access Control List Definitions (*mbxacl.conf*)	*$CMUHOME/med*
Auto Connect Definitions (ACD)	*$CMUHOME/acd*
Communication Ports Definitions (CPD)	*$CMUHOME/cpd*
Mailbox Control Definition (MCD)	*$CMUHOME/mcd*
Remote Site Definitions (RSD)	*$CMUHOME/rsd*
Security Protocol Definitions (SPD)	*$CMUHOME/spd*
Batch Encryption Definitions (*encrypt.cfg*)	*$CMUHOME/med*
Password Administration configuration file (*passadmpf.cfg*)	$CMUHOME/etc

Caution: Never edit the Mailbox Engines Definitions (MED) file manually. It is created and modified by execution of the **cmuinit** utility.
Never edit the AS2 configuration file directly. It is created at installation and modified using the Site Administration user interface.

2. Edit each file as necessary by adding, modifying, or deleting parameters. See *Editing the Batch Encryption Definitions File (encrypt.cfg)* on page 13 for information on editing that file. See Chapter 8, *Authentication Server Configuration*, for information on modifying the password administration configuration file (*passadmpf.cfg*).

Note: Definition file parameters *are not* case-sensitive in most cases. RSD and ACD file names *are* case-sensitive. Refer to Chapter 3, *Auto Connect Definitions*, and Chapter 6, *Remote Site Definitions*, for further details.

3. Save your changes.
4. Exit the file when it is complete.
5. Shut down Connect:Enterprise and restart to implement the changes.

Note: Some changes to the ACD may be implemented by using the **cmurefresh** utility without a shutdown and restart. Refer to Chapter 3, *Operator Commands*, in the *Connect:Enterprise UNIX User's Guide* for information on using **cmurefresh**.

Creating or editing an RSD file does not require a shutdown and restart.

Editing the Batch Encryption Definitions File (encrypt.cfg)

The Batch Encryption Definitions file contains information to support batch encryption on a per mailbox ID basis. The *encrypt.cfg* file lists mailbox IDs and indicates if batches added to the database using that mailbox ID are encrypted with strong, weak, 3DES, or no encryption. If a mailbox ID is not specified in *encrypt.cfg*, added batches are not encrypted. If *encrypt.cfg* does not exist or is empty, encryption is not performed on newly added batches for any mailbox ID.

Batch encryption ensures that information stored on the server is secure using standard, industry-strength algorithms. It is only available with Connect:Enterprise (Secure FTP). Encryption is turned on using the **ceukey** command described in Chapter 7, *Administrator Commands*, in the *Connect:Enterprise UNIX Installation and Administration Guide.* Only new batches added to the database with mailbox IDs specified in *encrypt.cfg* are encrypted. Batches stored in the database prior to batch encryption being activated are not encrypted.

Batch encryption does not interfere with any existing Connect:Enterprise functionality or with an existing Connect:Enterprise installation. Both encrypted and unencrypted batches are readable using the same means, and encryption is transparent to all users except through directory access to the batches.

Creating encrypt.cfg

To create *encrypt.cfg*, complete the following steps:

1. Open *$CMUHOME/med/sample_encrypt.cfg*.
2. Edit the file using the information in the following sections.
3. Save the file as *$CMUHOME/med/encrypt.cfg*.

Encryption Strength

The following table lists the possible encryption strengths that the system administrator can assign to individual mailbox IDs.

Encryption Strength	Definition
S|Strong	Encrypts incoming batches with a 24-byte key.
W|Weak	Encrypts incoming batches with an 8-byte key.
T| Triple_DES	Encrypts incoming batches with three 64-bit long keys (overall key length is 192 bits, although actual key length is 56 bits). Data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key. This makes 3DES three times slower than standard DES but offers much greater security.
N|None	Uses no encryption.

Syntax Example

The following is the *$CMUHOME/med/sample_encrypt.cfg* file. It contains samples of the information allowed in *encrypt.cfg*.

```
#Batch Encryption Definitions File
#
#Field 1 is the mailbox ID.  Field 2 is the encryption strength.
#All other fields are ignored and can be used for comments.
#
MboxIDa Triple_DES
MboxIDb T
MboxID1 S
MboxID2 Strong
MboxID3 W
MboxID4 Weak
MID*    N     #The only wildcard characters supported
MI?1    None  #are '*' and '?'
```

Guidelines

When modifying the *encrypt.cfg* file, observe the following guidelines:

- Comment lines begin with # and end with LF.
- The wildcard characters * and ? are supported in mailbox IDs.
- Any mailbox ID longer than 8 characters is ignored.
- If the first two fields (mailbox ID and encryption strength) are correct, the rest of the line is ignored.
- If a mailbox ID matches two or more mailbox ID specifications in the file, only the first one takes effect.

  For example:

```
MboxID* S
MboxID1 W
```

Because Mbox ID1 is addressed by MboxID*, all batches added for MboxID1 are added with strong encryption.

Priority of Values in RSD and ACD Definitions Files

Remote Site Definitions (RSD) files and Auto Connect Definitions (ACD) files are closely related and contain some of the same information about communications and data for a remote account that is authorized to access Connect:Enterprise. ACD files contain remote account-specific information in sections called remote blocks. An ACD file contains a remote section for each remote account that uses the auto connect schedule defined by the ACD. All RSD parameters that can also be set in the ACD file are optional in the ACD file. However, if the same parameters are set in both RSD and ACD files, the values in the ACD file override those specified in the RSD file. The RSD file and the ACD file are the only definitions files with such a relationship.

Chapter 2

Mailbox Access Control List Definitions (*mbxacl.conf*)

The /med directory contains a configurable mailbox access control list (ACL) file, called *mbxacl.conf*, to restrict access for mailbox ID security.

The *mbxacl.conf* file allows the system administrator to set up permissions on a mailbox-ID-by-mailbox-ID (not user-by-user) basis. The system administrator has all permissions in all mailbox IDs and determines the extent of the access for other mailbox IDs. All users have all permissions on their home mailbox ID. If *mbxacl.conf* does not exist, the default is for all users to have full access to all mailbox IDs.

The *mbxacl.conf* file specifies:

```
DEFAULT_ACL=permission [,...]
mailbox_id=permission[,permission...]
.
.
.
```

The first line establishes default permissions for all mailbox IDs that are not explicitly mentioned in the *mbxacl.conf* file. The subsequent lines establish permission sets for specific mailbox IDs.

The *mbxacl.conf* file is in effect only when the mailbox daemon (cmumboxd) is invoked with the **-r** parameter. Refer to the *Connect:Enterprise Installation and Administration Guide* for more information.

The mailbox daemon exits if **-r** is specified and it cannot locate the file or if the file contains a syntax error. If the *mbxacl.conf* file is missing or has a syntax error, the mailbox daemon will inform the Control daemon, which will then shut down all mailbox processes. Appropriate error messages will be displayed. Thus, the system administrator will not have to kill each process individually.

- The *mbxacl.conf* file is cached when the mailbox daemon starts (for improved performance), and whenever the file gets changed. The mailbox daemon will automatically re-cache the file when a mailbox ID is accessed (through commands such as **cmulist**, **cmudelete**, and so forth).

- Use the **cmulist** command to let the mailbox daemon re-cache the file after a change is made to the file. If a change is made during a remote session (after connecting to the mailbox daemon), the change will not affect the session. For example, if a remote issues a **dir** command while concurrently the system administrator changes the *mbxacl.conf* file and does a **cmulist** command, the remote command **dir** will not be affected by the changes in the file. Only ending and restarting the remote session will cause the updated file to be used.
- Re-caching takes effect only when the changes made to the *mbxacl.conf* do not introduce any syntax errors. If the new *mbxacl.conf* file has an error, the previously-cached version of the *mbxacl.conf* is still in effect and a standard error message is displayed.

Permissions

The following table lists all the possible permissions that the system administrator can assign to individual mailbox IDs. Generally, permissions that start with **F** indicate full access to all batches when using that particular command. Permissions that start with **R** indicate, when using that command, access is restricted to only the batches originated by the requesting user.

If two conflicting permissions are assigned (for example **RDIR** and **FDIR**), the more restrictive permission will be granted. In this case, it would be **RDIR**.

Permission	Definition	Explanation
ADD	Add	All users can add batches to this mailbox ID.
FDIR	Full Directory	All users can obtain an unrestricted directory listing of this mailbox ID, including all batches by all originators.
RDIR	Restricted Directory	All users can obtain a restricted directory listing of this mailbox ID, consisting only of the batches originated by the requesting user.
FDEL	Full Delete	All users can delete any batch in this mailbox ID, regardless of originator.
RDEL	Restricted Delete	All users can delete any batches in this mailbox ID that were originated by the requesting user. They cannot delete other batches originated by anyone else.
FREQ	Full Request	All users can request or extract any batches in the mailbox ID, regardless of originator.
RREQ	Restricted Request	All users can extract or request any batches in the mailbox ID that were originated by the requesting user.
FSTAT	Full Status	All users can obtain unrestricted status of the batches in the mailbox ID, regardless of originator.
RSTAT	Restricted Status	All users can obtain status of the batches in the mailbox ID that were originated by the requesting user.

Permission	Definition	Explanation
ALL	All	All users can perform all operations in the mailbox ID. This is equivalent to assigning the following commands: **ADD, FDIR, FDEL, FREQ, FSTAT.**
NONE	None	This mailbox ID is completely restricted to other users. Only the system administrator and the owner of the mailbox can perform operations. All other users cannot perform any functions on this mailbox ID.

Sample Syntax

The following is an example of the *mbxacl.conf* file that has been configured in the /med directory. It contains a default permission designation (the DEFAULT_ACL keyword), and permission sets for three other mailbox IDs.

```
//example mbxacl.conf file
DEFAULT_ACL= ADD, RDIR
mailbox_id1= ALL
mailbox_id2= ADD, RDIR, RSTAT,RDEL
mailbox_id3= NONE
```

The default applies to all mailbox IDs except mailbox_id1, mailbox_id2, and mailbox_id3. In this case, the users would be able to add batches to, and obtain restricted directory listings of all other mailbox IDs, because DEFAULT_ACL specifies **ADD** and **RDIR**.

The mailbox ID called mailbox_id1 would permit all users to perform all unrestricted operations (indicated by the **ALL** keyword). The mailbox ID called mailbox_id2 has a specific set of permissions that differs from the default. The third mailbox ID, mailbox_id3, is completely off limits to all users other than the system administrator and the user who logs in as mailbox_id3 (indicated by the **NONE** keyword).

Guidelines

When composing the *mbxacl.conf* file, observe the following guidelines:

- The access mailbox ID security permissions, such as **RDIR**, **FREQ**, and so forth are not case sensitive. However, the mailbox ID is case sensitive. For example, the following two lines would produce two different mailbox IDs.

```
id1 =Fdir, rreq, FSTAt
ID1 =FDIR
```

- Script style comments (for example, the line starts with the pound sign), C style comments (for example /* comment*/), or C++ style comments (for example, starts with //) are supported.

Setting Permissions for Mailboxes That Send and Receive AS2 Messages

If Mailbox Access Control is used, then all mailboxes used to receive or send AS2 messages must set permissions to allow all users access to (at a minimum) Directory, Status, Add, and Request functions. If this is not done, AS2 will not function properly.

You can set these permissions by defining DEFAULT_ACL in the mbxacl.conf file, which acts as a default for all mailboxes not explicitly defined, instead of defining the mailboxes individually.

In the configuration file, $CMUHOME/med/mbxacl.conf, set parameters for an individual mailbox or for all mailboxes. In the first example, which sets the permissions for a single mailbox, AS2MBOX is a sample mailbox name:

```
AS2MBOX = ADD, FDIR, FREQ, FSTAT
```

This example illustrates how to set default mailbox permissions for all mailboxes for which specific permissions have not been defined.

```
DEFAULT_ACL = ADD, FDIR, FREQ, FSTAT
```

Chapter 3

Auto Connect Definitions

The Auto Connect Definitions file defines an auto connect list that is either automatically initiated at predefined times or is consulted by the program during manual operation. Auto connect definitions (called schedules in the Connect:Enterprise UNIX Site Administration user interface), are stored individually—one auto connect, or schedule, per ACD file. The ACD file name is the auto connect name (sometimes referred to as the *auto connect list name*). In the Connect:Enterprise system, auto connect list names can be up to 15 characters; however, in reports only the first eight characters of the file name are displayed.

ACD files are also used as part of automatic routing feature. Specify **CONTACT=DATA_IMMEDIATE** in the ACD file if you intend to use automatic routing. Then add a batch using **$$ADD**, **cmuadd** with **-t** or **put** with **trigger=y**. Automatic routing sends files immediately to a destination upon receipt from other remotes or upon addition to the repository using the offline batch add utilities.

Both the **$$ADD** card and **cmuadd** contain the trigger parameter. Refer to the *Connect:Enterprise Remote User's Guide* and the *Connect:Enterprise Site Administration User Interface Help* for more information on the automatic routing feature.

The automatic routing feature also creates the autoroute file in the /med directory each time Connect:Enterprise UNIX is started with the **ceustartup** command. Refer to the *Automatic Routing* on page 46 for more information on the autoroute file.

The ACD file must have an extension of .acd in order to be recognized for automatic scheduling or automatic routing.

ACD File Conventions

ACD parameters can be specified in any sequence. Subordinate parameters associated with a major parameter can also be supplied in any sequence. White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Parameter names and keyword values are not case-sensitive.

The ACD filename *is* case-sensitive. In order for the **WHEN**, **EXCEPT**, and **CONTACT=DATA_IMMEDIATE** parameters to be parsed for a scheduled auto connect, the

ACD must be named using the *.acd* extension, lowercase. Use another extension for manual connects only.

Connect:Enterprise provides for a variety of remote site protocols within a given auto connect list. Any number and type of remote sites can be included in a single auto connect list definition. An overview of each ACD parameter follows.

Note: ACD parameters are not case-sensitive in most cases.

ACD Format

Required parameters are in bold in the following table. An overview of each parameter follows the table.

Parameters	Associated Values
CONTACT=	**ALWAYS|DATA|DATA_IMMEDIATE**
REMOTE=	**"***name***"**
ACPRIORITYLEVEL	*nn*
DISCINTV=	*nnnn*|NO
EXCEPT=	*crontab format,....*
INTERVAL=	*minutes*
REQUEUES=	*nnn*
RETRIES=	*nn*
SESSIONS=	*nnn*
STARTDELAY=	*seconds*
WHEN=	*crontab format,....*

Required Parameters

The following parameters, listed alphabetically, are required.

CONTACT=ALWAYS|DATA|DATA_IMMEDIATE

identifies when to contact the remote site. **DATA_IMMEDIATE** invokes the Automatic Routing feature that will send the data meeting the criteria to the remote site as soon as it appears in the mailbox IDs specified in the **SENDID** parameter. This parameter is only meaningful when **MODE=SENDONLY**. Otherwise, it will be ignored and the default will be used. When **CONTACT=DATA_IMMEDIATE**, the **WHEN** parameter in the ACD file is ignored.

Like full auto connects, ACD files specifying **CONTACT=DATA_IMMEDIATE** persist in the system even after their first execution.

Option	Description
ALWAYS	Contact the remote site regardless of whether there is data to send or not. This is the default.
DATA	Only contact the remote site if there is data to send.
DATA_IMMEDIATE	A **CONTACT=DATA_IMMEDIATE** ACD starts execution when batch matching the selection criteria specified in the ACD is available. The batch must be added with the trigger parameter through the **cmuadd**, **$$ADD**, or **put** command.

REMOTE="*name*"

identifies the name of a remote site to be contacted. More than one **REMOTE** parameter may be specified in a single ACD file; every **REMOTE** listed in the ACD will be processed when an auto connect occurs for the ACD. The names specified correspond to the like-named RSD files which must be defined in the ./rsd directory, with the exception of AS2. AS2 configuration parameters are specified in the AS2 configuration file. The **REMOTE** parameter is case-sensitive.

For the subparameters associated with the **REMOTE** block, refer to the Remote-Specific ACD Parameters section on *REMOTE-Specific Subparameters* on page 26.

Optional Parameters

The following parameters, listed alphabetically, are optional.

ACPRIORITYLEVEL=nn

specifies the order in which a connection request is queued waiting for connection resources to become available. Connection requests are ordered by the **ACPRIORITYLEVEL** setting, then by a first in, first out order that optimizes available communication resources. The priority range is 1–15 with 1 having the highest priority. The default is **07**.

DISCINTV=*nnnn*|NO

(Bisync only) specifies the number of seconds that Connect:Enterprise waits without session activity before the session with the remote site is terminated. When No is specified, it indicates that no disconnect processing is to be performed: the program waits the maximum line time-out period as stipulated by the protocol/remote site type before disconnecting. If a **DISCINTV** is not specified, the program uses the **DISCINTV** parameter from the RSD file. The default is **NO**. Valid values are 0-3600 and No.

EXCEPT=*crontab format, . . .*

specifies a crontab format similar to the one used with **WHEN**, but *excludes* time periods that are included by **WHEN**. **WHEN** is required when **EXCEPT** is used.

INTERVAL=*minutes*

identifies how many minutes Connect:Enterprise waits before it **REQUEUES** a remote site that it was unable to contact. **INTERVAL** does not postpone **RETRIES** done by the Communications daemon. The default is five minutes. Valid values are 0-120.

REQUEUES=*nnn*

indicates how many times one or more specified Resources will be requeued for any remote name in the list that is not connected after the number of retries specified in the **RETRIES** parameter have been attempted for each Resource. The total number of attempts made would be:

```
(# of Resources) x (RETRIES+1) x (REQUEUES+1).
```

For example, using the defaults, 5 requeues plus 1 retry with one resource would provide a total of 12 attempts. **REQUEUES** defaults to **5**. Valid values are 0–99.

If using the **REQUEUES** parameter with the **SESSIONS** parameter, refer to the description of the **SESSIONS** parameter for information on how to best optimize their use.

RETRIES=*nnn*

identifies how many attempts are made by a protocol daemon to contact an individual remote site. The default is one retry.

SESSIONS=*nnn*

specifies the maximum number of communications sessions permitted concurrently for a single auto connect. A value in the range of 0-128 is valid. A value of 0 is interpreted as maximum allowable sessions (128). Any value greater than 128 is truncated to 128. The **SESSIONS** parameter default value is **2**.

If all the remotes in an ACD must use a single resource, set **SESSIONS=1**. When the first remote is busy with the single resource, values greater than one permit the second and subsequent remotes to exhaust the specified requeue count and never get dialed.

If the number of available resources in an ACD is greater than one, but less than the number of remotes to be dialed by that ACD, set **SESSIONS** to the number of resources and **REQUEUES** to a nonzero value high enough to prevent the **REQUEUE** count from being exhausted. **STARTDELAY** and **INTERVAL** can both be increased as necessary to reduce the frequency of dial attempts.

STARTDELAY=*seconds*

specifies how many seconds the program waits after one session is completed before initiating another session. The **STARTDELAY** parameter is generally used to specify the time interval between the finish of one communications daemon session and the beginning

of another daemon session. The default is zero seconds; the next session starts immediately.

The **STARTDELAY** parameter is only valid when **SESSIONS**=**1**. If the value of **STARTDELAY** is greater than 0 (zero), Connect:Enterprise will force the value of **SESSIONS** to equal 1. Valid values are 0-180.

WHEN=*crontab format, ...*

Required for full auto connect. The **WHEN** parameter specifies the dates and times of day, using crontab formats, that the auto connect is run automatically. When auto connect is run automatically from this list it is called *a full auto connect.* If a **WHEN** parameter is not specified, you must initiate auto connects manually. Manual auto connects can be initiated using the **cmuconnect** command, whether or not **WHEN** is specified.

Crontab execution date/time specification uses five fields that are separated by spaces or tabs. They are integer patterns that specify the minute (0-59), hour (0-23), day of the month (1-31), month of the year (1-12), and day of the week (1-7, where 1=Sunday).

Each of these patterns may contain:

- A number in the respective range specified above.
- Two numbers separated by a minus, indicating an inclusive range.
- A list of numbers separated by commas, meaning all of these numbers.
- An asterisk, meaning all legal values.

For example:

```
WHEN = 30 12 * * 4
```

indicates that you want the auto connect to take place at 12:30 on every Wednesday during the entire year.

Several days and times can be specified in adjacent fields. For example:

```
0 0 1,15 * 2
```

would execute the auto connect on the first and fifteenth of each month and also on every Monday as indicated by the number 2 at the end. The two zeros at the beginning of the sequence indicates that the program should auto connect at midnight.

Another example:

```
0 0 * * 1
```

would run the auto connect on every Sunday.

REMOTE-Specific Subparameters

Associated with the **REMOTE** parameter in the ACD file are subparameters that define how the connection will be made. These subparameters can override the like-named RSD parameters for the remote account specified. Required parameters are in bold.

Parameter	Associated Value	Protocol
REMOTE=	"*name*"	All
ACRECVDIR=	*"pathname"*	FTP, SSHFTP
ACSENDDIR=	*"pathname"*	FTP, SSHFTP
ADDRESS= PHONE=	"*hostname*|*ipaddress*" "*phone number*", "*phone number*"	ALL
AUTOCONVERT=	ASCII|EBCDIC|NONE	Async, Bisync, FTP, BP
BATCHID	user batch ID	All
BCHSEP=	NONE|OPT1|OPT2|OPT3|OPT4|OPT5	Async, Bisync, FTP, SSHFTP
BLOCK=	*nnnn*	Bisync
BPNAME=	"*string*"	BP
CHECKHOSTIP=	YES|NO	SSHFTP
COMPRESSION=	YES|NO	Bisync, SSHFTP
CONCATETX=	YES|NO	Bisync
CONCATFILES	YES|NO	Async
DATAFORMAT=	ASCII|EBCDIC|BINARY (Async, FTP) ASCII|EBCDIC|BINARY (Bisync)	Async, Bisync, FTP
FTPPORT=	*nnnnn*	FTP, SSHFTP
FTP_PUT_OPTIONS=	"*string*"	FTP, SSHFTP, BP
GETCOMMAND	"*string*"	FTP, SSHFTP
LOGBATCH	YES|NO	Async, Bisync, FTP
MBXSEP=	YES|NO	FTP, SSHFTP
MODE=	SENDRECEIVE|SENDONLY|RECEIVESEND| RECEIVEONLY	Async, Bisync, FTP, SSHFTP
PARAMETERS	"*here*" document value	BP
PASSIVE	YES|NO	FTP

Parameter	Associated Value	Protocol
PASSWORDAUTH=	YES|NO	SSHFTP
POSTRECEIVE	"*string*"	FTP, SSHFTP
POSTSEND	"*string*"	FTP, SSHFTP
PORT_RANGE=	"nnnnn–nnnnn"	Async, Bisync, FTP
PORT_RETRIES=	nn	FTP
PORT_RETRY_WAIT_ TIME=	nnn	FTP
PREFERREDAUTH	"*password|publickey|password,publickey| publickey,password*"	SSHFTP
PRERECEIVE	"*string*"	FTP, SSHFTP
PRESEND	"*string*"	FTP, SSHFTP
PUBKEYAUTH=	YES|NO	SSHFTP
RECORDSEPARATOR=	CR|CRLF|LF (Async) CR|CRLF|LF|1E|1F (Bisync)	Async, Bisync
REMOTEFILENAME=	"*user_string*"	FTP, SSHFTP, Async with ZMODEM or KERMIT
RENAME_FILE=	YES|NO	FTP, SSHFTP
RESOURCE=	*daemon name:resource name,...daemon name: resource name,... (for async and bisync)* *daemon name (for FTP, SSHFTP, BP).*	All
SCAN=	YES|NO	Bisync
SECURITY_PROTOCOL_ FILE=	*/path/spd*	FTP
SENDID=	*name,name,name...*	Async, Bisync, FTP, SSHFTP
SESSIONEND	"*string*"	FTP, SSHFTP
SESSIONSTART	"*string*"	FTP, SSHFTP
SFI=	YES|NO	Bisync
SUNIQUE=	YES|NO	FTP, SSHFTP
TRUNC=	YES|NO	Bisync
TRUSTHOSTKEY	YES|NO	SSHFTP

REMOTE Optional Subparameters

The following parameters, listed alphabetically, are optional.

ACRECVDIR=*"pathname"*

(FTP, SSHFTP) specifies the remote site directory from which inbound files will be retrieved. If omitted, the value specified in the RSD is used instead. If the remote is a Connect:Enterprise site, this parameter is ignored.

For FTP, the pathname must be a full path name for the **cd** command to operate and the **TYPE** parameter in the RSD must be set to **REMOTE**.

ACSENDDIR=*"pathname"*

(FTP, SSHFTP) specifies the remote site directory to which outbound files will be sent. The pathname must be a full path name for the **cd** command to operate. For FTP, the **TYPE** parameter in the RSD must be set to **REMOTE**.

If omitted, the value specified in the RSD file is used instead. If the parameter is not specified in the RSD file either, Connect:Enterprise will try to send files to the home directory of the remote. If the remote is a Connect:Enterprise site, this parameter is ignored.

ADDRESS="*hostname|ipaddress*","*hostname|ipaddress*"
PHONE="*phone number*","*phone number*"

(All) specifies the telephone number used to contact the remote site. The **ADDRESS** parameter is used instead of **PHONE** to specify the IP address or host name for FTP remote sites. If the phone number contains embedded spaces, these must be placed in single or double quotes. This parameter accepts up to 128 characters.

Code every phone number and address that can be used to connect to the particular remote site. Multiple addresses are allowed. If omitted, the value in the RSD will be used instead. The value must be in double quotes.

During an auto connect, the communications daemon cycles through this list until a successful connection is made. Async connections wait for 45 seconds after dialing before attempting the next address. The amount of time that FTP connections wait before attempting the next address is system dependent.

Note: The **PHONE** parameter is ignored when the auto connect resource is a Bisync port defined as **LINE=LEASED**.

AUTOCONVERT=ASCII|EBCDIC|NONE

(All) identifies the data format in which the remote site wants to receive data.

Option	Description
ASCII	Data is sent as ASCII.
EBCDIC	Data is sent as EBCDIC fixed or variable length records. This causes all ASCII batches to be translated to EBCDIC. **Auto convert=ASCII** causes all EBCDIC batches to be converted. Binary batches will not be sent.
NONE	Data is sent in the same format as the source data. This is the default value.

If omitted, the value specified in the RSD is used instead.

BATCHID=

(All) specifies a user batch ID of 1–64 characters as the selection criteria. It must be enclosed in either single or double quotes if it contains spaces. Wildcard expressions are supported.

A pound sign followed by the batch number can be specified (for example, #14). The batch number can be up to 8 digits. Leading zeros are not required. One or more hyphenated ranges of batch numbers can be specified after the pound sign, separated by commas. Ranges can be mixed with unique batch numbers (for example, #57-59,95,100–110). The string must not exceed 64 characters including the pound sign.

BCHSEP= NONE|OPTION1|OPT1|OPTION2|OPT2|OPTION3|OPT3|OPTION4|OPT4|OPTION5|OPT5

(Async, Bisync, FTP, SSHFTP) specifies how Connect:Enterprise separates batches that are sent to a remote site. If omitted, the value specified in the RSD will be used instead.

Option	Description
NONE	Batches are not separated. If multiple batches are to be sent, they are sent as a single batch and are marked transmitted as the session progresses (see **Option3**). None is valid for all remotes (Bisync, FTP, Async, SSHFTP). This is the default. If **MBXSEP=YES**, all batches (even with different batch IDs) with the same mailbox ID will be concatenated into a file whose name is the mailbox ID. The batches belonging to separate mailbox IDs will be separated into files whose names match the corresponding mailbox IDs. The batches will be marked T (unless they are marked M) after they are transmitted. If **MBXSEP=NO**, Connect:Enterprise transmits all batches with the same batch ID as a single batch. All batches (even with different batch IDs) with the same or different mailbox IDs will be concatenated into a file whose name matches the remote user ID. The batches will be marked T (unless they are marked M) after they are transmitted.

Option	Description
OPTION1|OPT1	Connect:Enterprise uses the common RJE method of separating batches. At the end of each batch, Connect:Enterprise sends EOT, reads a response, and then sends ENQ to request use of the line. OPTION1 is valid for Bisync remote sites.
OPTION2|OPT2	Connect:Enterprise separates batches with an ETX. OPTION2 is valid for Bisync remote sites.
OPTION3|OPT3	Batches are not separated. If multiple batches are to be sent, they are sent as a single batch, however, the batches are not marked as transmitted until all the batches have been successfully transmitted. If **MBXSEP=YES**, all batches (even with different batch IDs) with the same mailbox ID will be concatenated into a file whose name is the mailbox ID. The batches belonging to separate mailbox IDs will be separated. The remote gets as many mailbox IDs as were specified in the Sendid list. The batches will be marked T (unless marked M) only after all the eligible batches have been transmitted successfully. If **MBXSEP=NO**, Connect:Enterprise transmits all batches with the same batch ID successfully before it flags any batches as transmitted. The batches are concatenated as a single file. All batches, even with different batch IDs, will be concatenated. Batches with different mailbox IDs will be concatenated into a file whose name matches the remote user ID. The remote receives one single file. Batches will be marked T (unless marked M) only after successful transmission of all the batches. **OPTION3** is valid for all remotes (Bisync, FTP, Async, SSHFTP).
OPTION4|OPT4	Batches are created as individual files at the remote site (in the current working directory) using the batch number assigned by Connect:Enterprise. The file name will follow the convention *batno.dat* (for example, 12345.dat). **OPTION4** is valid for FTP, Async, and SSHFTP remote sites. If **MBXSEP=YES**, all batches (even with the same batch IDs) within the same mailbox ID will be separated (for example to different filenames at the remote site). The batches belonging to the separate mailbox ID will be separated. The remote gets as many files as batches are eligible for transmission. The batches are marked T (unless marked M) after they are transmitted. If **MBXSEP=NO**, all batches (same or different batch IDs and mailbox IDs) will be separated. The batches are marked T (unless marked M) as they are transmitted. Each batch will be stored as a separate file, called batch_ID.batch_number if **RMT_FNAME_LEN=LONG** or batch_number.dat if **RMT_FNAME_LEN=SHORT**.
OPTION5|OPT5	If **MBXSEP=YES**, all batches with the same batch ID are concatenated into a file whose name is of the form mbx_id.bid where mbx_id is the mailbox ID and bid is the batch ID. All batches with different mailbox IDs or different batch IDs are separated. The batches are marked T (unless they are marked M) as they are transmitted. If **MBXSEP=NO**, all batches with the same batch ID, even across multiple mailboxes, are concatenated into one file. **OPTION5** is valid for FTP and SSHFTP remote sites.

BLOCK=*nnnn*

(Bisync) specifies the number of records per block for data outbound to a Bisync remote. The size of the buffer is controlled by the CPD file parameter, **SENDBUFF**. The default value is to fill the transmit buffer with as many logical records as can fit. The maximum value is **4096**.

If you specify a number of records per block that exceeds the capacity of the **SENDBUFF** parameter, Connect:Enterprise fills the buffer to capacity with as many whole logical records as can fit.

This parameter affects remote connects and auto connects for data outbound to Bisync remotes only. It can be overridden on remote connects by a similar parameter used on the **$$REQUEST** command. The **-B *nnn*** parameter of a **cmuconnect** command can override this parameter for both the ACD and RSD files.

BPNAME="*string*"

(BP) identifies the business process that is notified.

CHECKHOSTIP=YES|NO

(SSHFTP) Specifies to check the host IP address. This parameter is only valid if Connect:Enterprise is connecting to a SSHFTP server using a schedule (auto connect).

Yes = In addition to an identity check, SSH checks the host IP address in the known_hosts file. This allows SSH to detect if a host key changed due to DNS spoofing. Default.

No = No additional check is executed.

COMPRESSION=YES|NO

(Bisync, SSHFTP) This parameter is valid for Bisync and SSHFTP protocols, but operates differently for each as described in the following table:

Protocol	Description
Bisync	Identifies whether blank compression is applied to data that is outbound to the remote site. This parameter can be overridden by the **COMPRESSION** parameter in the ACD file. During transmission, Bisync compression reduces strings of three or more bytes of blanks (x ‘40’) to a compression indicator byte (x ‘1D’) and a length byte to indicate the number of blanks represented by the two bytes. A receiving station, on detecting the (x ‘1D’) compression indicator, reads the length byte that follows and decompresses the 2-byte string to the original string of blanks. Compression should not be requested for transparent batches. When blank compression and blank truncation are both enabled, truncation of trailing blanks is performed first, followed by blank compression of embedded strings of three or more blanks. **YES**—Requests Connect:Enterprise to compress blanks on data that is outbound to the remote site. **NO—**Indicates that outbound data will not be compressed. This is the default.

Protocol	Description
SSHFTP	Specifies whether to compress before SSH-2 encryption. YES = Batches are compressed at zlib level 6. NO = Batches are not compressed.

CONCATETX=YES|NO

(Bisync) controls how Connect:Enterprise processes inbound ETX-terminated Bisync batches.

NO—Inbound ETX-terminated batches will be added to the repository as individual batches. This is the default.

YES—Inbound ETX-terminated batches will be added to the repository as a single batch.

CONCATFILES=Y|YES|N|NO

(Async) specifies how Connect:Enterprise separates batches that are received from a site (inbound data). **CONCATFILES** is the parameter that deals with inbound data, whereas **BCHSEP** specifies how outbound data is treated.

The **CONCATFILES** parameter is contained in both the ACD and the RSD files. The value specified in the ACD file overrides the value specified in the RSD file. It is not available as an option on the **$$ADD** cards. Also, it is not available as a command line option to **cmuconnect** utility.

NO—each inbound file would be added as a separate batch. This is the default.

YES—each inbound file would be concatenated into one single batch.

- Use of **CONCATFILES** in Interactive Async Remote Connects

  If the Connect:Enterprise administrator has coded **CONCATFILES=NO** in the RSD for the remote, the selected files will be added as independent batches, each having the same mailbox id and batch id but different batch numbers. The mailbox id and batch id are taken from the **$$ADD** command that the remote entered at the command prompt (after supplying a loginid and password). For each batch that is added, the remote gets a directory record back at the end of the entire session.

  Inbound batch separation is accomplished with the help of the protocol and not by scanning the incoming data for embedded $$ cards. Connect:Enterprise does not scan for embedded $$ cards in Interactive mode, but the remote is given an opportunity to specify **$$ADD** options at the command line.

- Use of **CONCATFILES** in Non-Interactive Async Remote Connects

  In Non-Interactive mode, **CONCATFILES** is always assumed to be set to the default of NO. The value of **CONCATFILES** in the RSD has no effect. In this mode, remotes send their $$ commands inside of a file that they upload to Connect:Enterprise (also referred to as embedded $$ cards).

  Remotes are expected to sign on with /*SIGNON card or using **ID=** and **PASSWORD=** parameters on the $$ command. The remote then sends the desired

$$REQ, **$$DEL**, and **$$DIR** cards in the FIRST FILE before the first **$$ADD** card is coded. The Async server will look for a **$$ADD** card in the first 256 bytes or before the first linefeed character (whichever is earlier) of each subsequent file uploaded. The options on the previous **$$ADD** card will serve as default for the next **$$ADD** card. If no $$ cards are coded in any of the files uploaded and if the user has an RSD coded for the respective tty device, all the files will be added as separate batches. Also, unlike the Interactive mode, there will be no directory records sent to the remote, after each add.

Batch separation is accomplished with the help of protocol and not by scanning for embedded **$$ADD** cards. The scanning takes place at the beginning of each file after the Async server has received one **$$ADD** card. This is done to help the remote finish all its transmissions in one protocol session.

- Use of **CONCATFILES** in Async Auto Connect Sessions

  Inbound sessions during an auto connect behave like Non-Interactive Remote Connect Sessions except that the **CONCATFILES** parameter in the RSD is effective. During an auto connect, the Async server will scan for a **$$ADD** card in the first 256 bytes or until first linefeed character (whichever is first).

  Unlike the Non-Interactive sessions, no other $$ cards are valid even in the first file. For example, if the first file has a **$$REQ** card before any $$ cards or no $$ cards at all, the batch will be added with batch id = 'BATCH W/O $$ADD' and MBID = remote (as specified in **REMOTE=** parameter in the ACD file).

DATAFORMAT=ASCII|EBCDIC|BINARY

(Async, Bisync, FTP, SSHFTP) identifies the type of data that is received from the remote site.

Option	Description
ASCII	Data from the remote site consists of ASCII text strings. This is the default for Async and FTP.
EBCDIC	Data from the remote site consists of EBCDIC fixed or variable length records. This is the default for Bisync sites.
BINARY	Data from the remote site is binary or of unknown format.

If omitted, the value specified in the RSD is used instead.

FTPPORT=*nnnnn*

(FTP, SSHFTP) specifies the port number which is defined at the remote site for FTP operations. The default is **21**.

FTP_PUT_OPTIONS= “string”

(FTP, SSHFTP) designates one or more options (shown in the following table) that are automatically attached to a **put** command when transmitting a batch into the repository.

The string can be up to 256 characters, and must be surrounded by double quotes (for example, **FTP_PUT_OPTIONS="options..."**). **FTP_PUT_OPTIONS** can be specified in both ACD and RSD files; if specified in both, the specification in the ACD file overrides the specification in the RSD file, for auto connects.

There are three ways **FTP_PUT_OPTIONS** can be used:

- **Remote connects—**A remote FTP user places a batch into the repository with a standard-syntax (non-$$) **put** command. In this case, **ID** and **BID** are ignored.
- **Auto connects with receive mode enabled (that is, MODE=RECEIVEONLY or SENDRECEIVE or RECEIVESEND)**—In this case, the remote is a regular FTP server (does not need to be another Connect:Enterprise FTP server). Connect:Enterprise retrieves the files from the remote FTP server and places the batches in the repository with a mailbox ID specified by the ID option, if ID is specified in **FTP_PUT_OPTIONS**. If ID is not specified in **FTP_PUT_OPTIONS**, the batch(es) will be placed in the repository with a mailbox ID that matches the **REMOTE**. Either way, the added batch(es) will have the characteristics specified in **FTP_PUT_OPTIONS**.
- **Auto connects with send mode enabled (that is, MODE=SENDONLY or SENDRECEIVE or RECEIVESEND)**—In this case, the remote must be another Connect:Enterprise FTP server. Here, Connect:Enterprise will add the batch(es) into the remote repository with the specified options.

For example, an ACD file called xyz.acd contains the following:

```
REMOTE="abc"
MODE=SENDONLY
FTP_PUT_OPTIONS="MULTXMIT=Y"
```

No **SENDID** is specified, so the default **SENDID** is the **REMOTE** name (abc). The **cmuconnect** command is issued with xyz.acd as an argument; all requestable batches under the abc mailbox ID are sent to the remote repository with the M flag set (because of **FTP_PUT_OPTIONS**).

Note: **FTP_PUT_OPTIONS** can contain a batch ID specification (use **BID**); if you specify a batch ID containing spaces, the batch ID must be surrounded by single quotes that are preceded by backslashes. For example, **FTP_PUT_OPTIONS="BID=\’my batch\’"**

The following are the optional parameters that can be included in the **FTP_PUT_OPTIONS** string:

Parameter	Associated Value
BID=	\’xx...xx\’

Parameter	Associated Value
CODE=	A|E|B
EO=	YES|NO
ID=	XXXXXXXX
MULTXMIT=	YES|NO
PASSWORD=	XXXXXXXX
TO=	YES|NO
TRIGGER=	YES|NO
XMIT=	YES|NO

Optional Parameters for FTP_PUT_OPTIONS

The following parameters, listed alphabetically, are optional parameters when using the **FTP_PUT_OPTIONS**. In all cases, **YES** and **NO** can be abbreviated to **Y** and **N** respectively.

BID=\'xx...xx\'

identifies the 1–64 byte user batch ID for the batch being added. The entire phrase (**BID= \'xx...xx\'**) is required for $ syntax.

To create a single-word **BID**, use single quotation marks, each of which must immediately follow a backslash (\'xx.xx\'). To create a multi-word **BID** with spaces, use double quotation marks, each of which must immediately follow a backslash (\"xx...xx\"). The backslashes allow the Connect:Enterprise parser to process the single and double quote marks properly.

CODE=A|E|B

identifies the formats of the data being added. Three values are possible: **A** = ASCII, **E** = EBCDIC, and **B** = binary. The default is **A**.

EO=YES|NO

The **EO=Y** parameter specifies that this file can only be extracted once and never transmitted. If **YES** is entered, the batch is marked with a nontransmittable flag. Once extracted by the host site, it is also flagged unextractable. The default is **NO**.

ID=xxxxxxxx

identifies the 1–8 character mailbox ID assigned to your site.

MULTXMIT=YES|NO

enables or disables multiple transmissions of this batch. This parameter can be abbreviated to **MX**.

YES—Indicates that the batch can be transmitted multiple times. If **YES** is specified, it overrides the **XMIT** parameter and sets it to yes. With **MULTXMIT=Y**, the added batch is flagged as multi-transmittable at the time it is added to the mailbox ID, but unlike **XMIT=Y**, the batch is not flagged as transmitted when successfully transmitted. This leaves it eligible for subsequent transmissions that would not be possible if the transmitted flag were set.

NO—Indicates that the batch cannot be transmitted multiple times. This is the default.

PASSWORD=xxxxxxxx

enables a remote site to supply a password to the Connect:Enterprise Remote Command Exit for those sites that have enabled custom security.

TO=YES|NO

enables or disables a transmit once capability.

YES—Specifies that the batch can only be transmitted once. After transmission to the intended remote site, the batch is permanently locked and is flagged as nontransmittable and unextractable. If transmission of a batch with this parameter fails after one or more records have been transmitted, the batch is still locked. To retry the transmission, a new batch must be added from the original source.

NO—Specifies that the batch is not flagged as unextractable. This is the default.

TRIGGER=YES|NO

allows files to be rerouted immediately to other remotes. In order for automatic routing to function, an auto connect file must be defined with the **CONTACT=DATA_IMMEDIATE** parameter. When the characteristics of the batch being added match the selection criteria in this ACD file, the batch will automatically be forwarded to the destination specified in the ACD file.

YES—The batch will be rerouted if a valid auto connect list with matching selection criteria has been defined.

NO—The batch is not forwarded. This is the default.

XMIT=YES|NO

determines whether a batch is limited to host site use (in the network where the mailbox ID exists) or can be distributed to other locations.

YES—Specifies that the batch is available for transmission to any remote site which knows the proper mailbox ID. The batch is marked with a requestable flag. With

XMIT=Y, the added batch is flagged as transmitted after it is successfully forwarded to another remote.

NO—Specifies that the batch is available only for host site extraction. This indicates that the requestable flag is not set, restricting remote sites from requesting the batch. This is the default.

GETCOMMAND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent in place of the generic **MGET** command. A common use is to restrict the retrieve to specific files.

MBXSEP=YES|NO

(FTP, SSHFTP) dictates the behavior of the Connect:Enterprise FTP client for auto connects. It specifies how batches will be separated if they are coming from different mailbox IDs. This parameter works with the **BCHSEP** parameter.

MODE=SENDRECEIVE|SENDONLY|RECEIVESEND|RECEIVEONLY

(Async, SSHFTP, FTP) specifies the sequence in which Connect:Enterprise will communicate with remotes.

Option	Description
SENDRECEIVE	Connect:Enterprise first sends batches to the remote, then turns the line around to receive batches from the remote. This is the default.
SENDONLY	Connect:Enterprise sends batches to the remote, then disconnects from the remote site.
RECEIVESEND	Connect:Enterprise first receives batches from the remote site, then turns the line around to send batches to the remote.
RECEIVEONLY	Connect:Enterprise receives batches from the remote site, then disconnects from the remote site.

PARAMETERS= "*here*"

(BP) Specifies an XML "here" document value that is passed to the GIS business process. This document value must be constructed as follows:

```
PARAMETERS = <<EOF
Here is the value of the parameters parameter
It can be more than one line of info
EOF
```

The XML is not validated prior to notifying the BP.

PASSIVE=*Yes|No or True|False*

Specifies the FTP mode for the auto connect session. True or Yes indicates the FTP mode is passive. False or No means the FTP mode is active. False or No is the default.

PASSWORDAUTH=YES|NO

(SSHFTP) Specifies whether to use password authentication. This parameter is only valid if Connect:Enterprise is connecting to a SSHFTP server using a schedule (auto connect).

YES = The auto connect can perform password authentication.

NO = The auto connect does not perform password authentication.

PORT_RANGE=*"nnnnn–nnnnn"*

Specifies the pool of ports assigned for new socket operations. The string containing the numeric range or ranges must be enclosed in quotation marks. Separate multiple ranges with a comma within the string.

PORT_RETRIES=*nn*

Specifies the number of times the pool of ports is checked for an available port. The numeric value ranges from 0 to 99 with a default value of 0 (zero), or no retries.

PORT_RETRY_WAIT_TIME=*nnn*

Specifies the number of seconds to wait before the next attempt to connect to the port. The numeric value ranges from 0 to 180 with a default value of 0 (zero).

POSTRECEIVE="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately after the **GET** command.

POSTSEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately after the **PUT** command.

PREFERREDAUTH="*password;publickey;password,publickey, publickeypassword*"

(SSHFTP) Specifies the order in which the client prefers to attempt authentication methods. The following methods are available:

Method	Description
password	The remote will only use password authentication. You must set PASSWORDAUTH=YES.
publickey	The remote will only use public key authentication. You must set PUBKEYAUTH=YES. You must also create the following file and put the client's public key in the file: $CMUHOME/ssh/users/*account name*/authorized_keys.
password,publickey	The remote prefers password authentication. If password authentication is not available on Connect:Enterprise, or if password authentication fails, the client will use public key authentication. You must set PASSWORDAUTH=YES and PUBKEYAUTH=YES. You must also create the following file and put the client's public key in the file: $CMUHOME/ssh/users/*account name*/authorized_keys.
publickey,password	The remote prefers public authentication. If public key authentication is not available on Connect:Enterprise, or if public key authentication fails, the client will use password authentication. You must set PASSWORDAUTH=YES and PUBKEYAUTH=YES. You must also create the following file and put the client's public key in the file: $CMUHOME/ssh/users/*account name*/authorized_keys.

This parameter is only valid if Connect:Enterprise is connecting to a SSHFTP server using a schedule (auto connect).

PRERECEIVE="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the **GET** command.

PRESEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the **PUT** command.

PUBKEYAUTH=YES|NO

(SSHFTP) Specifies whether to attempt public key authentication. This parameter is only valid if Connect:Enterprise is connecting to a SSHFTP server using a schedule (auto connect).

Yes = The remote site can perform public key authentication.

No = The remote site cannot perform public key authentication.

RECORDSEPARATOR=CR|CRLF|LF|1E|1F

(Async, Bisync) identifies the end-of-line (EOL) character(s) to be used during the auto connect.

Option	Description
CR	Specifies that end-of-lines are delimited by a carriage return.
CRLF	Specifies that end-of-lines are delimited by a carriage return followed by a line feed.
LF	Specifies that end-of-lines are delimited by a line feed. This is the default for Async.
1E Bisync only	This parameter sets the record separator that will be used for transmission of nontransparent data to the remote site. This is the default for Bisync.
1F Bisync only	This parameter sets the record separator that will be used for transmission of nontransparent data to the remote site.

REMOTEFILENAME="*user_string*"

(FTP, SSHFTP, Async with ZMODEM or KERMIT) used as the destination file name on the remote system during FTP auto connects. This parameter can be no longer than 256 characters. Anything contained within the string will not be translated (for example **XMIT=Y** or anything similar). The string is simply forwarded. This string may not contain spaces, however all printable ASCII characters are allowed. If this parameter is coded in the RSD and the ACD, the value in the ACD file will take precedence.

RENAMEFILE|RENAME_FILE |REN_FILE=YES|Y|NO|N

(FTP, SSHFTP) provides for backward compatibility for FTP users using versions of Connect:Mailbox UNIX prior to version 3.1. If **YES** is chosen, whenever an FTP file transfer occurs, the previous Connect:Mailbox method will be retained. For example, the file XYZ is transferred to the remote's directory in a file called XYZ.tmp and then renamed XYZ.tmp to XYZ. If **YES** is chosen, the parameter **SUNIQUE** is ignored.

If **NO** is chosen, an FTP file transfer will occur without any renaming. Also, if **NO** is chosen, the value specified by **SUNIQUE** is valid and the remote user can determine whether or not to prevent files from being overwritten using that parameter.

RESOURCE=*daemon name:resource name,...*

(All) directs Connect:Enterprise to use a specific communications daemon, and a specific resource available to a communications daemon, in order to communicate with the remote site.

Caution: If you run more than one daemon (FTP, SSHFTP, async, or bisync) and it is important which daemon is used, you must specify a daemon in this field. For example, if you run two FTP daemons, a secure and a nonsecure daemon, and you want to use the secure daemon for auto connects, you must specify that daemon in this field.

The daemon name identifies the name of the communications daemon that will service the remote site. The resource name is optional and specifies the name of a resource the

daemon has access to, which will be used to communicate with the remote site. Multiple resource names separated by commas can be supplied for a single daemon name. Multiple daemon name:resource name sets can be separated by semicolons. FTP, SSHFTP, BP, HTTP, and EDIINT do not require resource names, only one or more daemon names. Bisync and Async make use of both daemon and resource names. To use a particular SSL-enabled FTP server instance for implicit SSL, specify the server name for the resource.

Note: This parameter is case-sensitive.

The resource name subparameters have different formats for the Async and Bisync daemons. These formats reflect the syntax of the Async and Bisync CPD file **DEVICE** and **PORT** keywords.

A sample async ACD resource parameter:

```
RESOURCE=ASYNC1: /dev/tty1, /dev/tty2
```

A sample ARTIC bisync ACD resource parameter:

```
RESOURCE=bisync1:c1p1,c1p2
```

A sample Cleo bisync ACD resource parameter:

```
RESOURCE=bisync:/dev/tty0
```

If this parameter is omitted, the **PROTOCOL** parameter in RSD is used to determine which communications daemon is used. If alternate routing is initiated, **RESOURCE** will not be used by the auto connect.

SCAN=YES|NO

(Bisync) controls how Connect:Enterprise will process inbound batches that contain embedded **$$ADD** cards.

NO—Embedded **$$ADD** cards will not be processed. They will be considered data.

YES—Embedded **$$ADD** cards will cause subsequent data to be added as a separate batch. This is the default.

Caution: Connect:Enterprise versions 1.1.01 and earlier behave as if **SCAN=NO** is enabled. Current users that want to add data to the repository (through the Bisync protocol) with embedded **$$ADD** cards for the express purpose of being sent to, and processed by, another Connect:Enterprise server must code an additional parameter (**SCAN=NO**) in their RSD files.

In the case where **SCAN=NO** is specified in the RSD file, **SCAN=YES $$ADD** card parameters will be honored only on physical batch boundaries. This is also true when a

SCAN=NO $$ADD card parameter is specified, regardless of the setting of the **SCAN** RSD parameter. After a **SCAN=NO $$ADD** card parameter has been processed, no more embedded **$$ADD** cards will be processed until the next physical batch boundary. A physical batch boundary is defined as the first inbound record of a session or the first inbound record following an ETX or EOT.

Scanning logic will be reset to the value specified in the RSD file at each ETX or EOT.

Note: When **SCAN=YES** is enabled, either through the **SCAN** RSD parameter or the **SCAN $$ADD** card parameter, embedded **$$ADD** cards will be scanned (sought). This will occur on every record in a nontransparent mode transfer and on the first record of every block in a transparent mode transfer.

SECURITY_PROTOCOL_FILE=*/path/spd*

(Secure FTP) specifies the name of the SPD file to be used during the FTP session. Refer to Chapter 7, *Security Protocol Definitions*, for more information regarding SPD files.

Note: This parameter is case-sensitive.

SENDID=name, name, name. . .

(Async, Bisync, FTP, SSHFTP) specifies the mailbox ID(s) of data that will be sent to the remote site and determines the sequence that data will be sent in.

Note: This parameter is case-sensitive.

If the **SENDID** is not specified, the batches that will be sent have IDs equal to the remote name (as specified for **REMOTE=** in the ACD file).

Note: If **MBXSEP=YES** and **BCHSEP=OPT5** for FTP, specify only one **SENDID**.

For example, consider the ACD file ACD1 as follows:

```
WHEN=30 12 * * 4
SESSIONS=1
INTERVAL=5
CONTACT=ALWAYS
REQUEUES=2

REMOTE="Acme"
   PHONE="5551212"
   MODE=SENDONLY
   SENDID=acme1, acme2

REMOTE="general"
   PHONE="5551213"
   MODE=SendOnly
```

Since **SENDID** is specified for the remote named acme; it will only be sent batches with ID=acme1 and ID=acme2.

However, **SENDID** is not specified for the remote named general. Therefore, it will be sent all eligible batches with the ID=general.

SESSIONEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the **QUIT** command.

SESSIONSTART="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during a FTP auto connect. The string is sent immediately after the **USER** and **PASS** commands.

SFI=YES|NO

(Bisync) suppresses the final IRS in each Bisync message block, producing a transmission with the message block formatted as in the two examples in the following figure:

```
02 - record - 26
02 - record - 1E - record - 1E - record - 26
```

If **SFI=NO**, the parameter constructs outbound Bisync message blocks with an IRS following each record in the block. This is the default. For example, an unblocked transmission (1 record per block) has message blocks formatted as in the following figure:

```
02 - record - 1E - 26
```

Blocked transmissions (more than one logical record per block) have message blocks formatted as in the following figure, where the block contains three records:

```
02 - record - 1E - record - 1E - record - 1E - 26
```

SUNIQUE=YES|NO

(FTP) This parameter is only meaningful when **RENAME_FILE=NO**.

YES—indicates that the file being transferred will not overwrite an existing file with the same name. If a file already exists with the same name, FTP will append .1 to the filename of the file being copied. Or, if a .1 file already exists, it will append .2 (and so on).

NO—may result in an existing same-named file being overwritten. This parameter must be specified in the **REMOTE** block of the .acd file.

TRUNC=YES|NO

(Bisync) identifies whether trailing blanks should be truncated before transmission to the remote site. The default is **NO**.

TRUSTHOSTKEY=YES|NO

(SSHFTP) If you are connecting to a SSHFTP server using a schedule (auto connect), and the SSHFTP server sends a host key, and that host key does not have a match in the known hosts file, this parameter controls whether to trust the host key sent.

If TRUSTHOSTKEY=YES, the host key provided is trusted. If there is a different host key in the known_hosts file, the known_hosts file is not updated. If there is no matching key in the known_hosts file, a key is added to the file. There can be separate entries in the known_host file for RSA and DSA key types. If TRUSTHOSTKEY= NO, the auto connect is rejected if there is not a matching key in the known_hosts file. If an auto connect is rejected for this reason, the only way the auto connect will be successful is to change the parameter to Yes or manually add the server's host key to the known_hosts file.

Sample ACD REMOTE Block

Four examples are given: *acme* is a Bisync site, *general* is an Async site, *national* is an FTP site, *northwest* is an SSHFTP. All sites will be contacted using the specified resource(s). Every keyword specifies a default if a default exits. Some parameters do not have defaults, so values have been supplied. Several of these remote-specific parameters can be specified in the RSD file for the remote. The ACD overrides RSD-specified values. Each of the following three remotes have been coded with every ACD parameter available for its resource type. Some are optional.

ASYNC, BISYNC and FTP are the default names for the Async, Bisync and FTP daemons. ASYNC1, ASYNC2, ASYNC3, BISYNC1 and FTP1 are the names of additional daemons that were started in this scenario. Their names would ordinarily be determined by issuing a **cmusession** command.

```
REMOTE="acme"
  AUTOCONVERT=NONE
  BCHSEP=NONE
  BLOCK=6
  COMPRESSION=NO
  DATAFORMAT=EBCDIC
  MODE=SENDRECEIVE
  PHONE="5551212"
  RESOURCE=bisyncd1:c1p1,c1p2,c2p1;bisyncd2:c1p1
  SENDID=acme1, acme2, acme3
  SFI=NO
  TRUNC=NO

REMOTE="general"
  AUTOCONVERT=NONE
  BCHSEP=NONE
  DATAFORMAT=ASCII
  MODE=SENDRECEIVE
  PHONE="5551213"
  RECORDSEPARATOR=LF
  RESOURCE=ASYNC:/dev/tty1,/dev/tty2;ASYNC1;ASYNC2:/dev/tty1;ASYNC3
  SENDID=acme1, acme2, acme3, acme4, acme5, acme6, acme7, acme8

REMOTE="national"
  ACRECVDIR=/home/ftp/download
  ACSENDDIR=/home/ftp/upload
  ADDRESS="999.999.999.999"
  AUTOCONVERT=NONE
  BCHSEP=NONE
  DATAFORMAT=ASCII
  FTPPORT=21
  FTP_PUT_OPTIONS="string"
  MBXSEP=NO
  MODE=SENDRECEIVE
  PASSIVE=YES
  PORT_RANGE="10024-10029"
  PORT_RETRIES=0
  PORT_RETRY_WAIT_TIME=0
  REMOTEFILENAME="user_string"
  RENAME_FILE=YES
  RESOURCE=FTP;FTP1
  SENDID=natla, natlb, natlc
```

```
REMOTE="northwest"
  ACRECVDIR=/home/sshftp/download
  ACSENDDIR=/home/sshftp/upload
  ADDRESS="888.888.888.888"
  AUTOCONVERT=NONE
  BCHSEP=NONE
  COMPRESSION = yes
  CIPHERS =
  "aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc"
  CHECKHOSTIP = YES
  DISCINTV = NO
  FTPPORT=22
  FTP_PUT_OPTIONS="string"
  MODE=SENDRECEIVE
  PORT_RETRIES=0
  PORT_RETRY_WAIT_TIME=0
  REMOTEFILENAME="user_string"
  RENAME_FILE=YES
  RESOURCE=SSHFTP
  SENDID=nwla, nwlb, nwlc
  MACS = "hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96"
  KNOWN_HOSTS = /ssh/known_hosts
  PASSWORDAUTH = Yes
  PORT_RETRIES=0
  PORT_RETRY_WAIT_TIME=30
  PREFERREDAUTH="publickey,password"
```

Automatic Routing

The automatic routing feature allows data to be sent to specified destinations as soon as it is placed in the repository (either from other remotes or through offline local commands). To use this feature, define **CONTACT=DATA_IMMEDIATE** in any ACD file (**CONTACT=DATA_IMMEDIATE** is only valid for **REMOTE**s that specify **MODE=SENDONLY**). The **WHEN** parameter in the ACD file is ignored when **CONTACT=DATA_IMMEDIATE** is defined.

Automatic routing is initiated whenever a **$$ADD** (Async/Bisync) or **put** ($$ FTP) command is issued with the trigger=y parameter, or a local offline **cmuadd** command is entered with the **-t** or **--trigger** parameter.

The *$CMUHOME/med/autoroute* file contains the names of the ACD files that are scanned for the matching criteria. The autoroute file only contains the names of ACD files that specify **CONTACT=DATA_IMMEDIATE**.

The autoroute file can be regenerated at any time with the **cmurefresh** command. For example the *$CMUHOME/med/autoroute* file contains the ACD filename x.acd. The ACD file called *$CMUHOME/acd/x.acd* contains:

```
CONTACT=DATA_IMMEDIATE
REMOTE=ABC
   SENDID=mboxid
   BATCHID=newid
   ADDRESS=my_node
   ACSENDDIR=/usr/tempdir
```

The system administrator enters local command:

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

The following events occur in this order:

1. file1, file2, and file3 are added as new batches to the mailbox ID mboxid. All three batches are added with batch ID newid.
2. The **-t** option in the **cmuadd** command triggers automatic routing.
3. Connect:Enterprise checks all ACD files listed in /med/autoroute for **REMOTE**s containing **CONTACT=DATA_IMMEDIATE**, a **SENDID** containing mboxid (or, if the **REMOTE**'s name is mboxid, **SENDID** is not specified), and a **BATCHID** of newid.
4. Since x.acd is listed in the autoroute file, Connect:Enterprise scans x.acd and finds **REMOTE=ABC** matching the criteria.
5. Connect:Enterprise automatically routes the three batches to the */usr/tempdir* directory on host my_node (per specification in **REMOTE=ABC**).
6. The batch separation method used for the three batches depends on the definition in the **REMOTE=ABC** section of x.acd, or, if not specified, the RSD file for remote ABC.
7. If more than one **REMOTE** was found matching the criteria from step 3 above, the process is repeated for each matching **REMOTE**.

ACD Files for AS2

When an AS2 schedule is created in the Connect:Enterprise Site Administration user interface, an ACD file is created for use by the EDIINT daemon. Connect:Enterprise then uses the EDIINT ACD file to create the HTTP ACD file to be used by the HTTP daemon. Both of these files have the same name, but the HTTP ACD begins with a period (.), so it is a hidden file. Additional ACD files are created during AS2 contract definition. One ACD files is created for each mailbox that remote site sends to. These additional ACD files are used for Async MDNs.

To ensure that the EDIINT and HTTP ACD files are always synchronized, they should only be edited through the Connect:Enterprise Site Administration user interface.

Caution: If you manually update either of these files, you must make the same updates to both files. The system automatically synchronizes these files *only* if you modify them from the Site Administration user interface.

The EDIINT ACD file is used by the EDIINT daemon to package the file to be sent according to the AS2 specification. After it is packaged, it is called an AS2 message. The AS2 message is then sent by the HTTP daemon to the remote partner using the HTTP ACD file.

Both the EDIINT and HTTP ACD files contain the following ACD parameters. Required parameters are in bold.

Parameters	Associated Values	Description Related to AS2
CONTACT=	ALWAYS|DATA| DATA_IMMEDIATE	For the EDIINT ACD file, this parameter can be any of the three available values. For the HTTP ACD file, this parameter is always DATA_IMMEDIATE, even if you manually change the value.
REMOTE=	**"***name***"**	For the EDIINT ACD file, this parameter will contain AS2 contract identifier information (sometimes called a "here" file). For the HTTP ACD file, this parameter is blank. The following characters need to be written in their escaped form: write & as & write < as < write > as > write " as "
SENDID=	*name,name,name...*	Specifies the mailbox ID(s) of data that will be sent. This parameter is case-sensitive. If it is not specified, AS2 messages will not be sent.
ACPRIORITYLEVEL	*nn*	
DISCINTV=	*nnnn*|NO	
EXCEPT=	*crontab format,....*	This parameter is not used in the HTTP ACD file.
INTERVAL=	*minutes*	
REQUEUES=	*nnn*	
RETRIES=	*nn*	
SESSIONS=	*nnn*	
STARTDELAY=	*seconds*	
WHEN=	*crontab format,....*	This parameter is not used in the HTTP ACD file.

In addition to the ACD parameters, both the EDIINT and HTTP ACD files contain the following remote block parameters. Required parameters are in bold.

Parameters	Values	Description
BATCHID	user batch ID	This field contains the batch ID (1-64 characters) or wildcard pattern that identifies what batches the EDIINT daemon processes. Enclose it in either single or double quotes if it contains spaces. Wildcard expressions are supported. A pound sign followed by the batch number can be specified (for example, #14). The batch number can be up to 8 digits. Leading zeros are not required. One or more hyphenated ranges of batch numbers can be specified after the pound sign, separated by commas. Ranges can be mixed with unique batch numbers (for example, #57–59,95,100–110). The string must not exceed 64 characters including the pound sign. **Note:** The HTTP ACD files contain wildcard batch IDs for packaged messages (*.RQ), for positive asynchronous MDNs (*.MD), and for negative asynchronous MDNs (*.NM). Do not change these values. The default value is *.PL. Do not set this value to asterisk (*), *.RQ, *.MD, or *.MN.
AS2URL	Specifies URL that the remote trading partner is using to receive AS2 messages.	If this URL is secure (HTTPS), the contract must specify appropriate SSL parameters. This OPTION must also be specified in the REMOTE block containing the BATCHID with value '*.RQ" pattern in the dot (.) or generated acd file.
AS2MIMETYPE	string	Specifies the type of file that you are sending. Valid values are: ◆ text ◆ application ◆ image ◆ audio ◆ video If AS2MIMETYPE is invalid, messages are sent with the type as application.

Parameters	Values	Description
AS2MIMESUBTYPE	string	Specifies the subtype of file that you are sending. You can define these values. They are not validated by Connect:Enterprise. Values included in the system are: ◆ plain—valid only for AS2MIMETYPE=text ◆ xml—valid only for AS2MIMETYPE=text ◆ edi-x12—valid only for AS2MIMETYPE=application ◆ edifact—valid only for AS2MIMETYPE=application ◆ octet-stream—valid only for AS2MIMETYPE=application If AS2MIMESUBTYPE is invalid, messages are sent with the type as octet-stream.
AS2COMPRESSION	TRUE|FALSE|NONE	Enables you to compress the message before transmission.
AS2SIGNING	YES|NO|NONE	Specifies whether to sign messages. If YES, you must specify AS2SIGNINGKEYCERT and AS2SIGNINGDIGEST.
AS2SIGNINGKEYCERT	filename	Specifies location of the private key certificate you use to sign messages.
AS2SIGNINGDIGEST	MD5|SHA1|NONE	Specifies signing algorithm your trading partner requires you to sign messages with.
AS2ENCRYPTION	YES|NO|NONE	Specifies whether to encrypt messages. If YES, you must specify AS2ENCRYPTIONCERT and AS2ENCRYPTIONALG.
AS2ENCRYPTIONCERT	filename	Specifies location of the public certificate your trading partner requires you to encrypt messages with.

Parameters	Values	Description
AS2ENCRYPTIONALG	3DES_168_CBC_PKCS5| DES_56_CBC_PKCS5| RC2_128_CBC_PKCS5| RC2_128_CBC_NONE| RC2_40_CBC_PKCS5| RC2_40_CBC_NONE| NONE	Specifies exchange algorithm your trading partner requires you to encrypt messages with.

Sample AS2 ACD Files

The following sample ACD file was created for EDIINT when a schedule was created:

```
CONTACT = DATA_IMMEDIATE
ACPRIORITYLEVEL = 7
DISCINTV = NO
INTERVAL = 5
REQUEUES = 0
RETRIES = 1
SESSIONS = 2
STARTDELAY = 0
REMOTE = <<EOF_MARKER
<id:AS2AccountIdentifier
  xmlns:id="http://www.sterlingcommerce.com/ceu/AS2/AccountIdentifier/v0.11"
  xmlns:conf="http://www.sterlingcommerce.com/ceu/AS2/Configuration/v0.11">
  <conf:MyAS2Identifier>local</conf:MyAS2Identifier>
  <conf:YourAS2Identifier>remote</conf:YourAS2Identifier>
</id:AS2AccountIdentifier>
EOF_MARKER
  AS2ENCRYPTIONCERT = "/path/to/exchange/cert"
  RESOURCE = EDIINT
  AS2MDNSIGNDIGEST = SHA1
  ##REMOTEDESCRIPTION AS2 Contract
  AS2SIGNING = YES
  AS2MIMESUBTYPE = "edi-x12"
  AS2URL = "http://remotehost.com:19980/as2/AS2IN/as2batch"
  AS2MIMETYPE = "application"
  BATCHID = "*.PL"
  AS2SIGNINGKEYCERT = "/path/to/signing/keycert"
  SENDID = AS2OUT
  AS2SIGNINGDIGEST = SHA1
  MODE = SendOnly
  AS2ENCRYPTIONALG = SYMMETRIC_ALG_3DES_168_CBC_PKCS5
  AS2ENCRYPTION = YES
  AS2COMPRESSION = TRUE
```

The following sample ACD file was created for HTTP when a schedule was created.

```
######################################################################
#
#  DO NOT MODIFY THIS FILE!
#
#  It has been automatically generated on Thu Mar 31 14:14:33 2005
#
######################################################################

CONTACT = DATA_IMMEDIATE
ACPRIORITYLEVEL = 7
INTERVAL = 5
REQUEUES = 0
RETRIES = 1
SESSIONS = 2
STARTDELAY = 0

REMOTE = <<EOF_MARKER
EOF_MARKER
  RESOURCE = HTTP
  SENDID = AS2OUT
  BATCHID = "*.RQ"
  MODE = SENDONLY
  AS2URL = "http://remotehost.com:19980/as2/AS2IN/as2batch"
```

The following sample ACD file was created for HTTP when defining an AS2 contract. It is used for sending asynchronous MDNs. It sends one MDN for each mailbox listed in the contract definition.

```
######################################################################
#
#  DO NOT MODIFY THIS FILE!
#
#  It has been automatically generated on Tue May 13 10:44:01 2003
#
######################################################################

CONTACT = DATA_IMMEDIATE
ACPRIORITYLEVEL = 7
INTERVAL = 5
REQUEUES = 0
RETRIES = 0
SESSIONS = 2
STARTDELAY = 0

REMOTE = <<EOF_MARKER
EOF_MARKER
  RESOURCE = HTTP
  SENDID = Orbiter
  BATCHID = "*.MD"
  MODE = SENDONLY
REMOTE = <<EOF_MARKER
EOF_MARKER
  RESOURCE = HTTP
  SENDID = Orbiter
  BATCHID = "*.NM"
  MODE = SENDONLY
```

Chapter 4

Communications Port Definitions

Connect:Enterprise can contain three different types of CPD files, each one supporting a different communications protocol. These files contain all the hardware and port options for their specific protocol, whether it is Bisync, Async, or FTP. These files are contained in the *./cpd* directory and are identified as cpd files by their file extension, *cpd*. For example, the Bisync CPD could be named *bisync.cpd* and the Async file could have the name *async.cpd*. There must be one CPD file defined for every communications daemon started.

Multiple communications daemons can be started, but each one must point to a unique CPD file, using the **-c** file name parameter.

Async CPD File

The Async CPD file is a free-form flat ASCII file which contains comments and port definitions. It is parsed at startup and any errors result in an error message and termination of the daemon.

The standard port settings for Async communications are 8 data bits, no parity and one stop bit (8-N-1).

A user that needs to connect with a value-added network or a legacy system using different parameters such as 7 data bits and even/odd parity can specify alternate parameters on a site-by-site basis.

Async CPD Conventions

Within a resource, Async CPD parameters can be specified in any sequence. Multiple configurations can be specified in the CPD. Comments can be placed between parameter values on separate lines or can be written after a parameters associated value. To mark off whole lines as comments, use a slash followed by an asterisk (/*) before the beginning of the comments and an asterisk followed by a slash (*/) at the end of the comments. Use a double forward slash (//) to mark all text after this point on the same line as a comment.

White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Comments cannot be placed between a parameter name and its associated value. Parameter names and keyword values are not case-sensitive in most cases.

Async CPD Format

The following table contains the Async CPD parameters and their definitions and usage. Required parameters are in bold.

Parameter	Associated Value
AUXPARMS=8|7,E|EVEN|O|ODD|N|NONE,1|2	Sets the number of bits, parity and stop bits for the serial ports. You must designate a choice for all three options. This parameter supports command interaction with the user in 8 or 7 bit, any parity, and 1 or 2 stop bit mode. However, all file transfers occur in 8 bit, no parity and 1 stop bit mode.
BAUD=1200|2400|4800|9600|19200|38400	Sets the rate at which Connect:Enterprise communicates with the modem, or the DTE rate. The DTE rate is the CONNECT speed (computer-to-modem). The maximum valid for the BAUD parameter on HP-UX machines is **19,200** and AIX machines permit **38,400**. The BAUD parameter has no effect on the DCE rates negotiated between modems at the time of connection. DCE rates are commonly referred to as the CARRIER speed (modem-to-modem).
COMMAND TIMEOUT=*nnn*	Determines the number of milliseconds of inactivity that can elapse before command input is considered *timed out*. The default value is **60** seconds. Valid values range from 0 to n. For example 60000 = 60 seconds If operating in noninteractive mode, the actual timeout will be double the COMMAND TIMEOUT.
DEVICE=*devicename*	Specifies a communications port to be serviced by the Async daemon. The device name is followed by optional device statements. A simple example of such a statement is: `device=/dev/tty1`
INTERACT=Y|YES|TRUE|N|NO|FALSE	Determines whether this port operates in interactive mode or noninteractive mode. Interactive mode is specified using **Y**, **YES**, or **TRUE**; noninteractive mode is specified using **N**, **NO**, or **FALSE**. The default is **YES**.
LOGON TIMEOUT=*nnn*	Specifies the number of milliseconds of inactivity that can elapse during each login prompt before a logon attempt is considered invalid and the session is terminated. The default value is **30** seconds. Valid values range from 0 to n, for example 60000 = 60 seconds.
PROTOCOL=XMODEM|YMODEM|ZMODEM|KERMIT|ASCII	Specifies which Async protocol the device should support at the beginning of a session. The specified value can later be overwritten in the RSD for a remote that uses that port. It can also be changed mid session in interactive mode using the **seta**, **setx, sety, setz, or setk** command at the command prompt, as explained in the *Connect:Enterprise Remote User's Guide.* The default is **ZMODEM**.

Parameter	Associated Value
ASCII_EOF_CHAR=*nnn*	Specifies the End of File character. The valid range is 0-255. This parameter is only valid if **PROTOCOL=ASCII**. If this parameter is set, the Async daemon sends the specified character after each batch or ends an incoming batch after receiving the character. By default, no EOF characters are recognized in the data stream and batch termination depends on the modem being disconnected or inactivity corresponding to the time specified in **ASCII_EOF_TIMEOUT**.
ASCII_EOF_TIMEOUT=*nnn*	Specifies the number of seconds of inactivity that can elapse before command input is considered *timed out*. The valid range is 0-300. The default is **0**. This parameter is required if **PROTOCOL=ASCII**.
FLOWCONTROL=TRUE|FALSE	Determines whether or not hardware flow control (RTS/CTS) is used during file transfers. Sites using Async protocols at high speeds must ask their remote sites to turn on the hardware flow control at their end. The default is **TRUE**. This parameter is required if **INTERACT=TRUE**.
LOGIN="*string*"	Defines the login prompt transmitted during initiation of an interactive session.
MDMCTL="*modem control string*"	Designates a modem initialization string. A colon denotes a send-expect pair. A semicolon terminates a send-expect pair.
PASSWD="*string*"	Defines the password prompt transmitted during initiation of an interactive session. This parameter is required if INTERACT=TRUE.
WAITDSR=YES|NO	Determines whether the Async daemon waits for the presence of a DSR signal from the device before sending the initialization string. The default is **YES**.

Sample Async CPD

The following figure shows the basic Async CPD file format.

```
DEVICE=/dev/tty4 {
    INTERACT=YES
    PROTOCOL=XMODEM
    BAUD=19200
    LOGON TIMEOUT=30000
    COMMAND TIMEOUT=60000
    MDMCTL="AT S0=1 E0 Q1 M1 &C1 &D2"
    WAITDSR=YES
    LOGIN="\rEnter login id: "
    PASSWD="\rEnter password: "
    AUXPARMS=7,O,2
}
DEVICE=/dev/tty2 {
    INTERACT=YES
    PROTOCOL=ZMODEM
    BAUD=19200
    LOGON TIMEOUT=30000
    COMMAND TIMEOUT=60000
    MDMCTL="AT S0=1 E0 Q1 M1 &C1 &D2"
    WAITDSR=Yes
    LOGIN="\rEnter login id: "
    PASSWD="\rEnter password: "
}
```

Bisync CPD File

Two types of Bisync CPD file define the available communications resources in the Bisync daemons. One is specific to ARTIC cards and one is specific to Cleo SYNCcable+ hardware. The specific commands and instructions for communicating between devices are contained in these daemons.

If you have Cleo SYNCcable+ hardware installed, the sample_cleo.cpd sample file, created during installation, contains examples of the required parameters. The Cleo bisync solution runs on all three major platforms that support Connect:Enterprise.

Bisync CPD Conventions

Bisync CPD parameters can be specified in any sequence. White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Parameter names and keyword values are not case-sensitive.

ARTIC Bisync CPD Format

The following table contains the ARTIC Bisync CPD parameters and their definitions and usage. Required parameters are in bold.

Parameter	Associated Value
PORT(card#,port#)=	Identifies the physical device that will be used for communications where *card#* is the card number and *port#* is the port number. Valid card numbers start at 0 and increase. Valid port numbers are 0–7. In a multi-board situation, define **PORT(0,0)**... **PORT(0,7),PORT(2,1)**... **PORT(2,7),PORT(3,1)**...**PORT(3,7)**, and so forth.
HARDWARE=MULTIPORT/2|PORTMASTER|MULTIPORTM2	Identifies the type of board a modem is attached to. The hardware type must be one of the following: 38-4-port SmartSync/DC 94-8-port SmartSync/DC. This is the default.
LINE=SWITCHED|LEASED	Specifies whether the data line is switched or leased (nonswitched). The default is switched. **Note:** When LINE=LEASED is specified as a Bisync CPD parameter, prior to a session, the cmusession command will display a State value of Connecting instead of Wait for Connect.
MODEM=*n*	Identifies the type of modem the Bisync daemon uses to communicate with the remote site. The default is **2**. The one-digit numeric modem type must be one of the following: 2-Other non-auto-dial external 3-External Racal-Vadic 4850PA (SADL) 8-External v.25bis compatible modems
801C=port:baud,databits, parity,stopbits	Defines an 801C ACU attached to an asynchronous port. The following parameters are used: port—system serial port to which the 801C ACU is attached. baud—baud rate (300, 1200, 2400, 4800, 9600, 19200, 38400). The default is 9600. databits—number of bits in a byte (7 or 8). The default is 8. parity—even, odd, or none (EVEN, ODD, or NONE). The default is NONE. stopbits—number of stopbits (1 or 2). The default is 1.

Parameter	Associated Value
MODEMCTRL=*string*	Specifies the modem-specific string. For the appropriate string to use with your modem, consult your modem documentation. The default is **"PRO2;1;1"**.

Examples of Modem Types

The following samples illustrate several modem configurations.

```
2-Other non-auto-dial external
UDS 201BC/AS Specifications:
  Non-Auto dialing
  4-wire, full-duplex leased lines OK
  2-wire, half-duplex leased lines OK
  2- or 4-wire half-duplex switched lines OK
  2400 bps Bell 201B/C (leased/switched)
UDS 208A/B Specifications:
  Non-Auto dialing
  4-wire, full-duplex leased lines OK
  2-wire, half-duplex leased lines OK
  2- or 4-wire half-duplex switched lines OK
  4800 bps Bell 208A/B

3-External Racal-Vadic 4850PA (SADL)
Racal-Vadic 4850PA Specifications:
  SADL Auto dialing
  No 4-wire, full-duplex leased line support
  2-wire, half-duplex leased line OK
  2- or 4-wire half-duplex switched lines OK
  4800 bps Bell 208
```

8-External v.25bis compatible modems
UDS 2140 201C Specifications:
v.25bis Auto dialing
No 4-wire, full-duplex leased line support
2-wire, half-duplex leased line OK
2- or 4-wire half-duplex switched line OK
2400 bps Bell 201C
2400 bps v.26bis
UDS 2860 208/201 Specifications:
v.25bis Auto dialing (enabled)
4-wire, full-duplex leased line OK
2-wire, half-duplex leased line OK
2- or 4-wire, half-duplex switched line OK
4800 bps Bell 208
2400 bps Bell 201B/C (leased/switched)
UDS v.3225 Specifications:
v.25bis Auto dialing (enabled)
(AT-Command set Auto dialing disabled)
4-wire, full-duplex leased line OK at 4800 and 9600 only
2-wire, half-duplex leased line OK
9600 bps trellis-coded v.32bis
9600 bps uncoded v.32bis
4800 bps uncoded v.32bis
2400 bps v.22bis (4-wire leased lines not supported)
300 bps Bell 103 (4-wire leased lines not supported)
UDS v.3229 Specifications:
v.25bis Auto dialing (enabled)
(AT-Command set Auto dialing disabled)
4-wire, full-duplex leased line OK
2-wire, half-duplex leased line OK
2- or 4-wire, half-duplex switched line OK
14400 bps trellis-coded v.32bis
12000 bps trellis-coded v.32bis
9600 bps trellis-coded v.32bis
9600 bps uncoded v.32bis
7200 bps trellis-coded v.32bis
4800 bps uncoded v.32bis
2400 bps v.22bis
1200 bps v.22bis
300 bps Bell 103

```
Codex 3260 v.32 Specifications:
  v.25bis Auto dialing (enabled)
  (AT-Command set Auto dialing disabled)
  No 4-wire, full-duplex leased line support
  2-wire, half-duplex leased line OK
  2- or 4-wire half-duplex switched line OK
  14400 bps trellis-coded v.32bis
  12000 bps trellis-coded v.32bis
  9600 bps trellis-coded v.32bis
  9600 bps uncoded v.32bis
  4800 bps uncoded v.32bis
  2400 bps v.22bis
  1200 bps v.22bis
  1200 bps Bell 212
  300 bps v.21
  300 bps Bell 103
```

Sample ARTIC Bisync CPD

The following sample shows the ARTIC Bisync CPD file format.

```
PORT(0,0){
    LINE=SWITCHED
    HARDWARE=MULTIPORT/2
    MODEM=2
}
PORT(0,1) = {
    LINE=SWITCHED
    HARDWARE=PORTMASTER
    MODEM=3
}

PORT(0,2)={
    LINE=SWITCHED
    HARDWARE=MULTIPORTM2
    MODEM=2
    801C=/dev/tty3:4800,7,E,1
}

PORT(1,0)={
    LINE=SWITCHED
    HARDWARE=MULTIPORT/2
    MODEM=2
    801C=/dev/tty9:300,8,none,1
}
```

Refer to the *Connect:Enterprise Installation and Administration Guide* for more information about configuring hardware and modem specifications.

Bisync CPD File Using Cleo SYNCcable+ Hardware and Cleo Bisync Daemon

In order to use SYNCcable+ devices with Connect:Enterprise, you must install and configure the hardware according to the instructions provided by Cleo. You must also configure Connect:Enterprise to use that hardware. A sample_cleo_cpd file is created during product installation.

Bisync CPD File Format Specific to Cleo SYNCcable+ Hardware

The following table contains parameters that are specific to Cleo SYNCcable+ connections, their definitions and usage. Required parameters are in bold.

Parameter	Description
CLEOINSTALL=/var/cleo	Specifies the full path to the system directory containing the Cleo SYNCcable+ software. The Cleo bisync daemon (cmubscdc) requires this path to locate the Cleo 3780Plus executable file and to run the 3780Plus process.
DEVICE=/dev/tty0	Identifies a SYNCcable+ device and specifies which async port is used for this device.
CLEOWORK=/var/cleo/tty0	Specifies a working directory for the Cleo 3780Plus process. This value must be unique for each device. Each 3780Plus process is required to have its own working directory, which becomes the current directory when the process is started. The working directory contains Cleo configuration files that override the files in the directory specified in the CLEOINSTALL path. The 3780Plus process creates files and places them in this directory, including the 3780.pid file that enables the Connect:Enterprise protocol daemon to locate the Cleo shared memory area used by the Cleo IAPI.

Parameter	Description
CLEOCMD=”-B 9600 -C testconfig”	Enables you to specify command line parameters used to set Cleo options when the 3780Plus process starts. The following command line parameters have defaults set in Connect:Enterprise that can be overridden when used with CLEOCMD=: ◆ -B Specifies the async port baud rate. The Connect:Enterprise default is 9600 if not coded otherwise using CLEOCMD=. Refer to the *3780Plus User’s Guide* for a list of valid values for this parameter. ◆ -A Instructs the 3780Plus process to enter IAPI mode immediately, and specifies the IAPI shared memory buffer size. The default memory buffer size is 20,000. Valid values range from 10000 to 20000 inclusive. The following command line parameters are valid with the SYNCcable+ device. Connect:Enterprise does not use them but will not negate them if used. ◆ -C Specifies the name of a Cleo configuration file that exists in the working directory. If not specified the 3780Plus process uses the default file, default.cfg, from the working directory. If the default does not exist in the working directory, the process runs with its own hard-coded default values. ◆ -L Specifies Cleo logging options. Cleo is capable of writing to the 3780.log file in the working directory. The following are valid logging option values: ◆ -LO Overwrites the log. ◆ -LA Appends to the log. ◆ -LN No logging. default Appends *only* if log already exists. ◆ -M Specifies a file name in the working directory for the serial line monitor output. This output captures the bisync protocol characters sent and received in a session and can be used for troubleshooting communications errors. The following command line parameters are ignored here because their values are fixed by Connect:Enterprise. ◆ -D Specifies the async port that the SYNCcable+ is connected to. ◆ -S Suppresses screen displays.

Parameter	Description
	The following command line parameters are ignored because they do not apply to the Connect:Enterprise environment. ◆ -A MSDOS base address value ◆ -E Set internal clocking ◆ -F Disables incoming file names ◆ -H Prevents line drop on hangup ◆ -I Set MSDOS IRQ ◆ -J Run an initial jobfile ◆ -K A jobfile option ◆ -N Suppress cr for printer files ◆ -R Another printer file option ◆ -T PRINT command option ◆ -U Another MSDOS parameter ◆ -W Prevents uncompressing 0x1d sequences
LINE=switched|leased	Specifies whether the line is switched or leased.
801C=/dev/tty2:9600,8,None,1	Specifies to use an 801C ACU to establish a dialup connection and provides details regarding how to access the ACU. This parameter parallels the ARTIC card bisync support. The value includes the async port name connected to the ACU, the baud rate, the parity setting, and the number of data bits and stop bits. The ACU is used for older and slower non-autodial modems such as the UDS 201 and UDS 208.

Sample Cleo Bisync CPD

The following sample .cpd file defines two SYNCcable+ devices:

```
CLEOINSTALL=/var/cleo

DEVICE=/dev/tty0 {
  CLEOWORK=/var/cleo/tty0
  CLEOCMD=”-B 9600 -C testconfig”
  LINE=switched
}
DEVICE=/dev/tty1 {
  CLEOWORK=/var/cleo/tty1
  CLEOCMD=”-M monitor1”
  LINE=switched
  801C=/dev/tty2:9600,8,None,1
}
```

FTP CPD File

The FTP file is a free-form flat ASCII file containing start-up options for FTP. It is parsed at start-up time and any errors will result in an error message and termination of the daemon.

FTP CPD Conventions

FTP CPD parameters can be specified in any sequence. White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Parameter names and keyword values are not case-sensitive.

FTP CPD Format

The following table contains the FTP CPD parameters and their definitions and usage. Required parameters are in bold.

Parameter	Definition
CLIENTPATHNAME=*/path/ftp*	Specifies the path and program name used to start the FTP auto connect server. The FTP program must be *ftp* for standard clients and *rftp* for clients using SOCKS.
PORTLISTENER=*nnnn*	Defines the port that remote sites use to access Connect:Enterprise. FTP client users use this port address to connect to Connect:Enterprise. The default is **10021**.
SERVERPATHNAME=*/path/ftpd*	Specifies the path and program name used to start the FTP server. The FTP program name must be *ftpd.*
DMZ_ADDRESS="*myhost.mydomain.com*"	Enables you to specify the address portion of the Passive response sent from the server to the client for a passive mode data connection or to specify the address portion of the Port command sent from the client to the remote server for an active mode auto connect. The string must be an IP address, a host name, or a fully qualified domain name.
KEEPALIVE=0|1|2|3	Specifies a keepalive value. This prevents the firewall from closing the control socket during transfers that are longer than the idle socket timeout value: ◆ 0—do not set keepalive for any sockets. ◆ 1—set keepalive only for control sockets used by autoconnect clients. ◆ 2—set keepalive only for control sockets established by remote connections. ◆ 3—set keepalive for both autoconnect clients and remote connections.

Parameter	Definition
MAILBOXMODEONLY=YES|NO	Specifies the mode the FTP server is to use. Connect:Enterprise can completely replace the functions of FTP on the UNIX system as well as perform Connect:Enterprise-specific FTP functions. **YES**—Indicates that Connect:Enterprise only allows Connect:Enterprise operations. In **MAILBOXMODEONLY** mode, the Connect:Enterprise FTP server does not allow any operation outside of Connect:Enterprise. This is the default. **NO—**Allows Connect:Enterprise to perform standard FTP functionality in addition to Connect:Enterprise operations. The FTP daemon must be started with root authority when this option is used.
PASSIVE=YES|NO	Indicates whether the FTP client will attempt to connect to a remote server in passive mode in order to navigate SOCKS firewalls. The default is **NO**.
PORT_RANGE=*"lllll–hhhhh"*	Specifies the pool of ports assigned for new socket operations. Use up to five numeric ranges where the format "*lllll-hhhhh*" represents the beginning and end of each range. The low element of the range denoted as 'lllll' must be a number greater or equal to 1025. The high element of the range denoted as 'hhhhh' must be a number less than or equal to 65535. The low element must be numerically less than the high number. The string must be enclosed in quotation marks.
PORT_RETRIES=*nn*	Specifies the number of times the pool of ports is checked for an available port. The numeric value ranges from 0 to 99 with a default value of 0 (zero), or no retries. Setting this parameter in the CPD file defines defaults that can be overridden for one or more connections by settings in the ACD file or the RSD file.
PORT_RETRY_WAIT_TIME=*nnn*	Specifies the number of seconds to wait before the next attempt to connect to the port. The numeric value ranges from 0 to 180 with a default value of 0 (zero). Setting this parameter in the CPD file defines defaults that can be overridden for one or more connections by settings in the ACD file or the RSD file.
PORTRESTRICTION=YES|NO	Indicates whether the FTP client will restrict connections to port numbers above 1024 and the IP address of the data channel will always match that of the control channel. Setting this restriction may cause problems for users navigating through SOCKS firewalls, but may prevent certain types of security attacks.

Parameter	Definition
SECURITY_PROTOCOL_FILE=*/path/spd*	(Secure FTP only) specifies the name of the SPD file to be used during the FTP session. If **SECURITY_PROTOCOL_FILE** is not specified, Secure FTP is disabled for remote connects, regardless of Secure FTP parameters specified elsewhere. The SPD file name must be a valid path on the host containing the FTP communications daemons. This parameter will override the default of *$CMUHOME/spd/ssl.spd*. Refer to Chapter 7, *Security Protocol Definitions* for more information regarding SPD files.
USE_DNS=TRUE|FALSE	Enables or disables the use of DNS for logging and tracing.

Sample FTP CPD

The following is a sample FTP CPD file. The path names will vary depending on your system type.

```
PortListener=10021
ServerPathname=/users/mailbox/cmunix/aix/bin/ftpd
ClientPathname=/users/mailbox/cmunix/aix/bin/ftp
MailboxModeOnly=Yes
```

SSHFTP CPD File

The SSHFTP CPD file is a free-form flat ASCII file containing start-up options for SSHFTP. It is parsed at start-up time and any errors will result in an error message and termination of the daemon. For SSHFTP, the CPD file defines the port definition for inbound SSHFTP connections.

SSHFTP CPD Conventions

SSHFTP CPD parameters can be specified in any sequence. White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Parameter names and keyword values are not case-sensitive.

SSHFTP CPD Format

The following table contains the SSHFTP CPD parameters and their definitions and usage. Required parameters are in bold.

Parameter	Definition
CLIENTPATHNAME=*/path/ftp*	Specifies the path and program name used to start the SSHFTP server. The SSHFTP program must be sshd.

Parameter	Definition
PORTLISTENER=*nnnn*	Defines the port that remote sites use to access Connect:Enterprise. SSHFTP client users use this port address to connect to Connect:Enterprise. The default is **10022**.
SERVERPATHNAME=*/path/ftpd*	Specifies the path and program name used to start the SSHFTP server. The SSHFTP program name must be *sshftpd.*
COMPRESSION = YES|NO	Specified whether compression is allowed from a remote client.
CIPHERS = "*cipher1,cipher2, . . .*"	Specifies the ciphers allowed for encryption in order of preference. Separate multiple ciphers with commas. The default is: aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc. This parameter does not apply when you are connecting to a SSHFTP server using a schedule (auto connect).
MACS = "*MAC1, MAC2, ...*"	Specifies the message authentication code (MAC) algorithms in order of preference. The MAC algorithm is used for data integrity protection. Separate multiple algorithms with commas. The default is: hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96. This parameter does not apply when you are connecting to a SSHFTP server using a schedule (auto connect).
PASSWORDAUTH = YES|NO	Specifies whether Connect:Enterprise performs password authentication for SSH. You can also use the long form, PASSWORDAUTHENTICATION.
PUBKEYAUTH = YES|NO	Specifies whether Connect:Enterprise performs public key authentication for SSH. You can also use the long form, PUBKEYAUTHENTICATION.
SCPALLOWED=YES|NO	Specifies whether to accept commands from an SSH SCP client. The default is NO.
USEDNS = YES|NO	Specifies whether the SSH daemon (cmusshd) should look up the remote host name and check that the resolved host name for the remote IP address maps back to the same IP address.

Sample SSHFTP CPD

The following is a sample SSHFTP CPD file. The path names will vary depending on your system type.

```
PORTLISTENER = 12233
SERVERPATHNAME  = /sci/users/mailbox/ceunix/bin/sshd
CLIENTPATHNAME  = /users/mailbox/cmunix/aix/bin/sshftp
PasswordAuth = Yes
PubKeyAuth = Yes
Compression = Yes
UseDns = Yes
```

Chapter 5

Mailbox Control Definitions

The Mailbox Control Definitions (MCD) file contains the Connect:Enterprise system-level parameters. This includes such definitions as the:

- System name
- Security parameters
- Exit enable/disable flags
- Automatic log file maintenance parameters

While reconfiguring Connect:Enterprise, the administrator should rarely need to change the basic definitions.

MCD Conventions

MCD parameters can be specified in any sequence. Subparameters associated with a parameter can also be supplied in any sequence. White space lines between parameters are ignored. Multiple parameters can be placed on a single line, separated by spaces or tabs. Parameters cannot span multiple lines. Parameter names and keyword values are not case-sensitive.

MCD Format

The required parameters are in bold in the following table.

Parameter	Associated Value
SYSTEM=	**name**
APIFUNCTIONEXIT=	**YES|NO**
BATCHRECEIVEEXIT64=	**YES|NO**

Parameter	Associated Value
BATCHRECEIVEEXIT=	**YES|NO**
BATCHSENDEXIT64=	**YES|NO**
BATCHSENDEXIT=	**YES|NO**
INITIALIZATIONEXIT=	**YES|NO**
LOGEXIT=	**YES|NO**
REMOTECOMMANDEXIT=	**YES|NO**
SECURITYEXIT=	**YES|NO**
SESSIONINITEXIT=	**YES|NO**
SESSIONTERMEXIT=	**YES|NO**
SESSINITBUFFEXIT=	**YES|NO**
TERMINATIONEXIT=	**YES|NO**
MAXLOGFILES=	*nn*
MAXLOGFILESIZE=	*nnnnn*
POLLDAEMONS=	*seconds*
SECURITY=	BATCH|NONE
SIPSENCRYPTION	**YES|NO**
VALIDIDLIST=	*ID,ID,ID...*
EXTERNALAUTH	No|Yes|RSD
EXTERNALAUTHPORT	*port*
EXTERNALAUTHSECUREPORT	*port*
EXTERNALAUTHCONTROLPORT	*port*
EXTERNALAUTHRESOURCE	*access definition name*
EXTERNALAUTHHOST	*host name*
EXTERNALAUTHPORT	*port*
EXTERNALAUTHSPD	*fully qualified path to SPD*

Required Parameters

The following parameter is required.

SYSTEM=*name*

identifies the Connect:Enterprise system name. The name can be up to 8 characters. **SYSTEM=" "** is also valid.

Exits

The Exit parameters are required for *control.mcd*. The MCD identifies which exits within the system are activated. When an exit point is activated, a user-defined subroutine is called and passed the appropriate information. For more information about these exits, refer to the *Connect:Enterprise Programmer's Guide*. An overview of each exit activation parameter follows in alphabetical order.

APIFUNCTIONEXIT=YES|NO

identifies whether the API invokes an exit before an operation requested by a Connect:Enterprise command line utility or user-written API is performed. When No is specified, it indicates that an API function exit is not being used. The default is **NO**.

BATCHRECEIVEEXIT64=YES|NO

identifies whether the mailbox daemon invokes an exit when a batch of data is:

- Received from a remote site
- Added by the **cmuadd** utility
- Added by an API

When No is specified, it indicates that a Batch Receive exit is not being used. The default is **NO**.

BATCHRECEIVEEXIT=YES|NO

identifies whether the mailbox daemon invokes an exit when a batch of data is:

- Received from a remote site
- Added by the **cmuadd** utility
- Added by an API

When No is specified, it indicates that a Batch Receive exit is not being used. The default is **NO**. Compatible only with batches smaller than 2,147,483,647 bytes. For compatibility with batches of 2,147,483,647 or more bytes, use BATCHRECEIVEEXIT64.

Note: BATCHRECEIVEEXIT and BATCHRECEIVEEXIT64 are mutually exclusive.

BATCHSENDEXIT64=YES|NO

identifies whether a communications daemon invokes an exit when a batch of data has been successfully sent to a remote site. When No is specified, it indicates that a Batch Send exit is not being used. The default is **NO**.

BATCHSENDEXIT=YES|NO

identifies whether a communications daemon invokes an exit when a batch of data has been successfully sent to a remote site. When No is specified, it indicates that a Batch Send exit is not being used. The default is **NO**. Compatible only with batches smaller than 2,147,483,647 bytes. For compatibility with batches of 2,147,483,647 or more bytes, use BATCHSENDEXIT64.

Note: BATCHSENDEXIT and BATCHSENDEXIT64 are mutually exclusive.

INITIALIZATIONEXIT=YES|NO

identifies whether the Control daemon invokes an exit as part of its initialization processing. When No is specified, it indicates that an Initialization exit is not being used. The default is **NO**.

LOGEXIT=YES|NO

identifies whether the Log daemon invokes an exit before a record is written to the Log File. When No is specified, it indicates that a Log exit is not being used. The default is **NO**.

REMOTECOMMANDEXIT=YES|NO

identifies whether a communications daemon invokes an exit before a command requested by a remote site is executed. When No is specified, it indicates that a Remote Command exit is not being used. The default is **NO**.

SECURITYEXIT=YES|NO

invoked before access to a mailbox ID is given. Validates the user and password information.

SESSIONINITEXIT=YES|NO

identifies whether or not a communications daemon invokes an exit when a session is established with a remote site. When No is specified, it indicates that a Session Initiation exit is not being used. The default is **NO**.

SESSIONTERMEXIT=YES|NO

identifies whether a communications daemon invokes an exit before communications with the remote site are terminated. When No is specified, it indicates that a Session Termination exit is not being used. The default is **NO**.

SESSINITBUFFEXIT=YES|NO

identifies whether a communication daemon running in noninteractive mode invokes an exit when it receives the first block of data from the remote site. The default is **NO**.

SIPSENCRYPTION=YES|NO

identifies whether internal product communications are encrypted. You must create the encryption key using **cmusipskey** utility. Refer to the *Administrator Command* chapter of the *Connect:Enterprise Installation and Administration Guide* for information on the **cmusipskey** utility. The default is **NO**.

TERMINATIONEXIT=YES|NO

identifies whether the Control daemon invokes an exit during Connect:Enterprise system shutdown. When No is specified, it indicates that a Termination exit is not being used. The default is **NO**.

Optional Parameters

The following parameters, listed alphabetically, are optional.

MAXLOGFILES=*nn*

identifies the number of log files that will be stored before they are deleted. This parameter accommodates a value up to 64. The default is **2**.

MAXLOGFILESIZE=*nnnnn*

identifies the maximum size a log file can become in kilobytes. Valid values are 2048–32000 kilobytes. The default is 2048 kilobytes.

POLLDAEMONS=*seconds*

specifies how frequently (in seconds) the Control Daemon should poll the other Online Connect:Enterprise daemons to see if they are still responding to processing requests. The default specifies that daemons are polled every **300** seconds (5-minute intervals).

SECURITY=BATCH|NONE

limits inbound batches to those added from a remote site with a valid mailbox ID. The valid mailbox IDs are identified with one or more **VALIDIDLIST** parameters. The default is **NONE**.

VALIDIDLIST=*ID, ID, ID...*

lists valid mailbox IDs for **SECURITY=BATCH** and requires the **SECURITY=BATCH** parameter. Mailbox IDs, separated by commas, can be up to 8 characters each and the entire line can be up to 1,024 bytes.

EXTERNALAUTH

Specifies whether external authentication or Connect:Enterprise authentication is used.

No = default Use Connect:Enterprise authentication.

Yes = Use external authentication.

RSD = Authentication method is determined by the account definition.

EXTERNALAUTHPORT

Specifies the port that the external authentication service listens on.

You can specify any port between 1–65535. The default is 61365.

EXTERNALAUTHSECUREPORT

Specifies the SSL port that the external authentication service listens on.

You can specify any port between 1–65535. The default is 61366.

EXTERNALAUTHCONTROLPORT

Specifies the port that Connect:Enterprise uses to send the passphrase to the external

You can specify any port between 1–65535. The default is 61367.authentication service.

EXTERNALAUTHRESOURCE

Specifies the name of a default server access definition or server group on the external authentication service.

You can specify any valid access definition or server group name.

EXTERNALAUTHHOST

Specifies the host that the external authentication service runs on.

You can specify any valid host name. The default is CMUHOST.

EXTERNALAUTHPORT

Specifies the fully qualified path name of the executable for the external authentication service.You can specify any valid path.

EXTERNALAUTHSPD

Specifies the fully qualified path to the security definition that contains the SSL keys required for secure connection to the external authentication server. This is only required if SSL is used.You can specify any valid security definition path.

Sample MCD File

The example below shows an MCD file with values defined.

```
SYSTEM=“Mailbox”
POLLDAEMONS=300
MAXLOGFILESIZE=2048
SIPSENCRYPTION=NO
SECURITY=BATCH
VALIDIDLIST=acme, general, allied, national
INITIALIZATIONEXIT=NO
TERMINATIONEXIT=NO
SESSIONINITEXIT=YES
SESSIONTERMEXIT=NO
REMOTECOMMANDEXIT=NO
APIFUNCTIONEXIT=NO
BATCHRECEIVEEXIT=YES
BATCHRECEIVEEXIT64=NO
BATCHSENDEXIT=NO
BATCHSENDEXIT64=NO
LOGEXIT=NO
SESSINITBUFFEXIT=NO
SECURITYEXIT=NO
```

Chapter 6

Remote Site Definitions

Remote Site Definitions files (RSD) identify remote sites (accounts) that are authorized to access a Connect:Enterprise system. They also contain operational characteristics for each remote site.

RSDs may also define a password for a local site user. An RSD file is required for each local site user that uses passwords.

If your remote site has the capability of supporting multiple protocols, you must create an RSD file for each protocol that you can use (even if it is your emergency option).

RSDs are stored in the *$CMUHOME*/rsd directory under the name of the remote site or user's system login ID.

Types of RSD Files

There are two types of RSD files:

- RSDs that define remote site accounts
- RSDs that are password files for local site Connect:Enterprise users

RSD files are parsed by the Control Daemon when a communications session is started with a remote site. RSDs are also parsed when command line utilities are executed by local site users in order to match passwords.

Remote Account RSD

For remote site RSDs, the name of the RSD file indicates the remote site for which the connection is defined. These are the same names that will be specified by remotes when they login for remote connects. These names are also specified within the ACD files, on the **REMOTE** statements, that identify the remotes to be contacted during an auto connect. The filenames must be 1–8 characters in length, alphanumeric.

Local User RSD

Connect:Enterprise command line utilities require a password to be matched against the user's system userid. These files must be named with the same value a user logs in to the UNIX host with. Each contains a single record that reads: **PASSWORD="password"**.

The specified password should not match the user's system password, but should instead be a unique password that will be used to match the value a user specifies with the -p switch on all the command line utilities.

For example: A user with the userid of USER1 would execute the command **cmulist -p mypasswd** to match an RSD file named USER1 containing the statement **PASSWORD="mypasswd"**.

It is imperative that the entire ./rsd directory have permissions set to 0700 (Owner Read/Write/Execute). Use the UNIX **chgrp** command to modify the group names of the executables in the ./bin directory. Group your users so that sensitive command line utilities such as **cmushutdown, cmuinit, cmufixup**, and **cmurebuild** will not be available to every local site user. For more information, refer to your UNIX operating system's manuals.

Several of these remote-specific parameters can also be specified in the **REMOTE** sections of an ACD file that lists the remotes for an auto connect. All RSD parameters that can also be coded in the ACD are optional in the ACD. However, if coded in both places, the ACD values override the RSD values. In some cases, the RSD parameters also supply default parameters for **$$ADD** and **$$REQUEST** cards.

RSD Conventions

When you create an RSD file, adhere to the following conventions:

- Specify RSD parameters in any sequence.
  - Specify subordinate parameters associated with a major parameter in any sequence.
- Place each parameter on a single line.
  - Do not span multiple lines with a single parameter.
  - White space lines between parameters will be ignored.
- Parameters and keywords are not case-sensitive with the exception of the password value and the RSD file name.

RSD Format

The following tables show the contents and format of RSD files.

Local RSD Format

The only parameter required for local users is password. Refer to the following table.

Parameter	Associated Value
PASSWORD=	**"***string***"**
TRANSLATE=	"*filename*"
LOGBATCH=	YES|NO

Remote RSD Format

All parameters (including **PASSWORD**) are valid for the remote sites. Required parameters are bold in the following table. An overview of each RSD parameter follows the table.

Note: The PASSWORD parameter is required with Connect:Enterprise (Secure FTP).

Parameter	Associated Value	Protocol
PROTOCOL=	BSC|BISYNC|FTP|XMODEM|YMODEM|ZMODEM| KERMIT|ASCII|SSHFTP|BP	All
ACRECVDIR=	*"pathname"*	FTP, SSHFTP
ACSENDDIR=	*"pathname"*	FTP, SSHFTP
ADDRESS= PHONE=	"*hostname*|*ipaddress*" "*phone number*","*phone number*"	All
ASCII_EOF_CHAR=	*nnn*	Async
ASCII_EOF_TIMEOUT=	*nnn*	Async
AUTOCONVERT=	ASCII|EBCDIC|NONE	Async, Bisync, FTP
BCHSEP=	NONE|OPT1|OPT2|OPT3|OPT4|OPT5	Async, Bisync, FTP, SSHFTP
BLOCK=	*nnnn*	Bisync
CHECKHOSTIP=	YES|NO	SSHFTP
CIPHERS=	"*cipher1,cipher2,...*"	SSHFTP
COMPRESSION=	YES|NO	Bisync, SSHFTP
CONCATETX=	YES|NO	Bisync

Parameter	Associated Value	Protocol
CONCATFILES=	Y|YES|N|NO	Async
DATAFORMAT=	ASCII|EBCDIC|BINARY (Async, FTP) ASCII|EBCDIC|BINARY (Bisync)	Async, FTP, Bisync
DIRFORM=	MVS|UNIX|CLIENT|BROWSER	Async, Bisync, FTP
DISTINGUISHEDNAME=	"*string*"	All
DISCINTV=	*nnnn*|NO	Bisync
EXTERNALAUTH=	YES|NO|NEVER	All
EXTERNALAUTHRESOURCE=	*"string"*	All
FTP_PUT_OPTIONS=	"*string*"	FTP, SSHFTP, BP
FTPPORT=	*nnnnn*	FTP, SSHFTP
FTPSCRIPT=	"*ipaddress,portno,[loginid,password]*", "*ipaddress,portno,[loginid,password]*",...	FTP
GETCOMMAND=	"*string*"	FTP, SSHFTP
LOGBATCH=	YES|NO	Async, Bisync, FTP, SSHFTP
LOGONMSG=	"80-char */SIGNON string used to connect to a JES2 remote type"	Bisync
LOGOFFMSG=	"80-char */SIGNOFF card string used to sign off from a JES2 remote type"	Bisync
MACS=	"*MAC1,MAC2,...*"	SSHFTP
MAILBOX_LIST=	"*mailbox_ID,...*"|"*"	FTP, SSHFTP
MBXSEP=	YES|NO	FTP, SSHFTP
MODE=	SENDRECEIVE|SENDONLY|RECEIVESEND| RECEIVEONLY	Async, Bisync, FTP, SSHFTP, BP
PASSWORD=	"*password*"	All
PASSWORDAUTH=	YES|NO	SSHFTP
PORT_RANGE=	"nnnnn–nnnnn"	FTP
PORT_RETRIES=	nn	FTP
PORT_RETRY_WAIT_ TIME=	nnn	FTP
POSTRECEIVE=	"*string*"	FTP, SSHFTP

Parameter	Associated Value	Protocol
POSTSEND=	"*string*"	FTP, SSHFTP
PREFERREDAUTH=	"*password*|*publickey*| *password,publickey,publickey,password*"	SSHFTP
PRERECEIVE=	"*string*"	FTP, SSHFTP
PRESEND=	"*string*"	FTP, SSHFTP
PUBKEYAUTH=	YES|NO	SSHFTP
RECORDSEPARATOR=	CR|CRLF|LF|1E|1F (Async) CR|CRLF|LF|1E|1F (Bisync)	Async, Bisync
REMOTEFILENAME=	"*string*"	FTP, SSHFTP, Async with ZMODEM or KERMIT
RENAME_FILE=	YES|NO	FTP, SSHFTP
RMT_FNAME_LEN=	SHORT|LONG	FTP, SSHFTP
RMT_PASSWORD=	"*password*"	FTP, SSHFTP, BP
RMT_USER=	"*username*"	FTP, SSHFTP, BP
SECURITY_PROTOCOL _FILE=	*/path/spd*	FTP
SCAN=	YES|NO	Bisync
SCRIPT=	BEGIN...END	Async
SENDBUFF=	*nnnn*	Bisync
SESSIONSTART=	"*string*"	FTP, SSHFTP
SESSIONEND=	"*string*"	FTP, SSHFTP
SFI=	YES|NO	Bisync
TRANSLATE=	"*filename*"	Async, Bisync, FTP
TRUNC=	YES|NO	Bisync
TRUSTHOSTKEY	YES|NO	SSHFTP
TYPE=	REMOTE|CMUMBOX|JES2 (Bisync)	Bisync, FTP

Required Parameters

The following parameters, listed alphabetically, are required.

PROTOCOL=BSC|BISYNC|FTP|SSHFTP|BP|XMODEM|YMODEM|ZMODEM|ASCII|KERMIT

Identifies how the remote site and Connect:Enterprise communicate with each other. If the remote account is on a Gentran Integration Suite (GIS) server, specify BP.

Optional Parameters

The following parameters, listed alphabetically, are optional.

ADDRESS="*hostname|ipaddress*","*hostname|ipaddress*" PHONE="*phone numbe*r","*phone number*"

PHONE—(Async, Bisync) specifies the telephone number used to contact the remote site. The **ADDRESS** parameter is used instead of **PHONE** to specify the IP address or host name for FTP remote sites. This parameter accepts 1–127 characters.

During an auto connect, the communications daemon cycles through this list until a successful connection is made. Async connections wait for 45 seconds after dialing before attempting the next address. The amount of time that FTP connections wait before attempting the next address is system dependent.

Note: The **PHONE** parameter is ignored when the auto connect resource is a Bisync port defined as **LINE=LEASED**.

ACRECVDIR="*pathname*"

(FTP, SSHFTP) Specifies the remote site directory from which inbound files will be retrieved. This parameter can be overridden using the **ACRCVDIR** parameter in the ACD file.

The path name must be a full path name for the **cd** command to operate. Also, the **TYPE** parameter in the RSD must be set to **REMOTE**. If the remote is a Connect:Enterprise site, this parameter is ignored.

ACSENDDIR="*pathname*"

(FTP, SSHFTP) specifies the remote site directory to which outbound files will be sent. This parameter can be overridden using the **ACSENDDIR** parameter in the ACD file. If the remote is a Connect:Enterprise site, this parameter is ignored.

The path name must be a full path name for the **cd** command to operate. Also, the **TYPE** parameter in the RSD must be set to **REMOTE**. If the remote is a Connect:Enterprise site, this parameter is ignored.

ASCII_EOF_CHAR=*nnn*

(Async) specifies the End of File character. The valid range is 0–255. This parameter is only valid if **PROTOCOL=ASCII**.

If this parameter is set, the async daemon sends the specified character after each batch, or ends an incoming batch after receiving the character. By default, no EOF characters are recognized in the data stream, and batch termination depends on the modem being disconnected or inactivity corresponding to the time specified in **ASCII_EOF_TIMEOUT**.

ASCII_EOF_TIMEOUT=*nnn*

(Async) specifies the number of seconds of inactivity that can elapse before command input is considered *timed out*. The valid range is 0–300. The default is **0**.

This parameter is required if **PROTOCOL=ASCII**.

AUTOCONVERT=ASCII|EBCDIC|NONE

(Async, Bisync, FTP, SSHFTP, BP) identifies the type of data format in which the remote site wants to receive.

Option	Description
ASCII	Data is sent as ASCII text.
EBCDIC	Data is sent as EBCDIC fixed or variable length records. This causes all ASCII batches to be translated to EBCDIC. Bisync remotes do not have to specify the CONV=E parameter on their **$$REQUEST** commands when making remote connection to Connect:Enterprise. Instead, edit the appropriate RSD file to add AUTOCONVERT=EBCDIC.
NONE	The data is sent in the same format as the source data. This is the default value.

This parameter can be overridden by the ACD file.

BCHSEP=NONE|OPTION1|OPT1|OPTION2|OPT2|OPTION3|OPT3|OPTION4|OPT4|OPTION5|OPT5

(Async, Bisync, FTP, SSHFTP, BP) specifies how Connect:Enterprise separates batches that are sent to a remote site (outbound data). This parameter can be overridden by the **BCHSEP** parameter in the ACD file.

Option	Description
NONE	Batches are not separated. If multiple batches are to be sent, they are sent as a single batch and are marked transmitted as the session progresses (see **OPTION3**). **NONE** is valid for all remotes (Bisync, FTP, and Async). This is the default. If **MBXSEP=YES**, all batches (even with different batch IDs) with the same mailbox ID will be concatenated into a file whose name is the mailbox ID. The batches belonging to separate mailbox IDs will be separated into files whose names match the corresponding mailbox IDs. The batches will be marked T (unless they are marked M) after they are transmitted. If **MBXSEP=NO**, Connect:Enterprise transmits all batches with the same batch ID as a single batch. All batches (even with different batch IDs) with the same or different mailbox IDs will be concatenated into a file whose name matches the remote user ID. The batches will be marked T (unless they are marked M) after they are transmitted.
OPTION1|OPT1	Connect:Enterprise uses the common RJE method of separating batches. At the end of each batch, Connect:Enterprise sends EOT, reads a response, and then sends ENQ to request use of the line. **OPTION1** is valid for Bisync remote sites.
OPTION2|OPT2	Connect:Enterprise separates batches with an ETX. **OPTION2** is valid for Bisync remote sites.
OPTION3|OPT3	Batches are not separated. If multiple batches are to be sent, they are sent as a single batch, however, the batches are not marked as transmitted until all the batches have been successfully transmitted. If **MBXSEP=YES**, all batches (even with different batch IDs) with the same mailbox ID will be concatenated into a file whose name is the mailbox ID. The batches belonging to separate mailbox IDs will be separated. The remote gets as many mailbox IDs as were specified in the Sendid list. The batches will be marked T (unless marked M) only after all the eligible batches have been transmitted successfully. If **MBXSEP=NO**, Connect:Enterprise transmits all batches with the same batch ID successfully before it flags any batches as transmitted. The batches are concatenated as a single file. All batches, even with different batch IDs, will be concatenated. Batches with different mailbox IDs will be concatenated into a file whose name matches the remote user ID. The remote receives one single file. Batches will be marked T (unless marked M) only after successful transmission of all the batches. **OPTION3** is valid for all remotes (Bisync, FTP, and Async).

Option	Description
OPTION4|OPT4	Batches are created as individual files at the remote site (in the current working directory) using the batch number assigned by Connect:Enterprise. The file name will follow the convention *batno.dat* (for example, 12345.dat). **OPTION4** is valid for FTP and Async remote sites. If **MBXSEP=YES**, all batches (even with the same batch IDs) within the same mailbox ID will be separated (for example to different filenames at the remote site). The batches belonging to the separate mailbox ID will be separated. The remote gets as many files as batches are eligible for transmission. The batches are marked T (unless marked M) after they are transmitted. If **MBXSEP=NO**, all batches (same or different batch IDs and mailbox IDs) will be separated. The batches are marked T (unless marked M) as they are transmitted. Each batch will be stored as a separate file, called batch_ID.batch_number if **RMT_FNAME_LEN=LONG** or batch_number.dat if **RMT_FNAME_LEN=SHORT**.
OPTION5|OPT5	If **MBXSEP=YES**, all batches with the same batch ID are concatenated into a file whose name is of the form mbx_id.bid where mbx_id is the mailbox ID and bid is the batch ID. All batches with different mailbox IDs or different batch IDs are separated. The batches are marked T (unless they are marked M) as they are transmitted. If **MBXSEP=NO**, all batches with the same batch ID, even across multiple mailboxes, are concatenated into one file. **OPTION5** is valid for FTP remote sites.

BLOCK=*nnnn*

(Bisync) specifies the number of records per block for data outbound to a Bisync remote. The size of the buffer is controlled by the CPD file parameter, **SENDBUFF**. The default value is to fill the transmit buffer with as many logical records as can fit. The maximum value is 4096.

If you specify a number of records per block that exceeds the capacity of the **SENDBUFF** parameter, Connect:Enterprise fills the buffer to capacity with as many whole logical records as can fit.

This parameter affects remote connects and auto connects for data outbound to Bisync remotes only. It can be overridden on remote connects by a similar parameter used on the **$$REQUEST** command. The **-B *nnn*** parameter of a **cmuconnect** command can override this parameter for both the ACD and RSD files.

CHECKHOSTIP=YES|NO

(SSHFTP) Specifies to check the host IP address.

Yes = In addition to an identity check, SSH checks the host IP address in the known_hosts file. This allows SSH to detect if a host key changed due to DNS spoofing. Default.

No = No additional check is executed.

CIPHERS= "*cipher suite1, cipher suite 2, . . .*"

(SSHFTP) Specifies the ciphers allowed for encryption in order of preference. Separate multiple ciphers with commas. The default is:

```
aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc
```

COMPRESSION=YES|NO

(Bisync, SSHFTP) This parameter is valid for Bisync and SSHFTP protocols, but operates differently for each as described in the following table:

Protocol	Description
Bisync	Identifies whether blank compression is applied to data that is outbound to the remote site. This parameter can be overridden by the **COMPRESSION** parameter in the ACD file. During transmission, Bisync compression reduces strings of three or more bytes of blanks (x '40') to a compression indicator byte (x '1D') and a length byte to indicate the number of blanks represented by the two bytes. A receiving station, on detecting the (x '1D') compression indicator, reads the length byte that follows and decompresses the 2-byte string to the original string of blanks. Compression should not be requested for transparent batches. When blank compression and blank truncation are both enabled, truncation of trailing blanks is performed first, followed by blank compression of embedded strings of three or more blanks. **YES**—Requests Connect:Enterprise to compress blanks on data that is outbound to the remote site. **NO—**Indicates that outbound data will not be compressed. This is the default.
SSHFTP	Specifies whether to compress before SSH-2 encryption. **YES** = Batches are compressed at zlib level 6. Default. **NO** = Batches are not compressed.

CONCATETX=YES|NO

(Bisync) controls how Connect:Enterprise processes inbound ETX-terminated Bisync batches.

NO—Inbound ETX terminated batches will be added to the repository as individual batches. This is the default.

YES—Inbound ETX terminated batches will be added to the repository as a single batch.

CONCATFILES=Y|YES|N|NO

(Async) specifies how Connect:Enterprise separates batches that are received from a site (inbound data). **CONCATFILES** is the parameter that deals with inbound data, whereas **BCHSEP** specifies how outbound data is treated.

The **CONCATFILES** parameter is contained in both the ACD and the RSD files. The value specified in the ACD file overrides the value specified in the RSD file. It is not available as an option on the **$$ADD** cards. Also, it is not available as a command line option to **cmuconnect** utility.

NO—each inbound file would be added as a separate batch. This is the default.

YES—each inbound file would be concatenated into one single batch.

- Use of **CONCATFILES** in Interactive Async Remote Connects

  If the Connect:Enterprise administrator has coded **CONCATFILES=NO** in the RSD for the remote, the selected files will be added as independent batches, each having the same id and batch id but different batch numbers. The id and batch id are taken from the **$$ADD** command that the remote entered at the command prompt (after supplying a loginid and password). For each batch that is added, the remote gets a directory record back at the end of the entire session.

  Inbound batch separation is accomplished with the help of the protocol and not by scanning the incoming data for embedded $$ cards. Connect:Enterprise does not scan for embedded $$ cards in Interactive mode, but the remote is given an opportunity to specify **$$ADD** options at the command line.

- Use of **CONCATFILES** in Non-Interactive Async Remote Connects

  In Non-Interactive mode, **CONCATFILES** is always assumed to be set to the default of NO. The value of **CONCATFILES** in the RSD has no effect. In this mode, remotes send their $$ commands inside of a file that they upload to Connect:Enterprise (also referred to as embedded $$ cards).

  Remotes are expected to sign on with /*SIGNON card or using **ID=** and **PASSWORD=** parameters on the $$ command. The remote then sends the desired **$$REQ**, **$$DEL**, and **$$DIR** cards in the FIRST FILE before the first **$$ADD** card is coded. The Async server will look for a **$$ADD** card in the first 256 bytes or before the first linefeed character (whichever is earlier) of each subsequent file uploaded. The options on the previous **$$ADD** card will serve as default for the next **$$ADD** card. If no $$ cards are coded in any of the files uploaded and if the user has an RSD coded for the respective tty device, all the files will be added as separate batches. Also, unlike the Interactive mode, there will be no directory records sent to the remote, after each add.

  Batch separation is accomplished with the help of protocol and not by scanning for embedded **$$ADD** cards. The scanning takes place at the beginning of each file after the Async server has received one **$$ADD** card. This is done to help the remote finish all its transmissions in one protocol session.

- Use of **CONCATFILES** in Async Auto Connect Sessions

  Inbound sessions during an auto connect behave like Non-Interactive Remote Connect Sessions except that the **CONCATFILES** parameter in the RSD is effective. During an auto connect, the Async server will scan for a **$$ADD** card in the first 256 bytes or until first linefeed character (whichever is first).

  Unlike the Non-Interactive sessions, no other $$ cards are valid even in the first file. For example, if the first file has a **$$REQ** card before any $$ cards or no $$ cards at all, the batch will be added with batch id = ’BATCH W/O $$ADD’ and id = remote (as specified in **REMOTE=** parameter in the ACD file).

DATAFORMAT=ASCII|EBCDIC|BINARY

(Async, Bisync, FTP) identifies the type of data that is received from the remote site. **DATAFORMAT** can be overridden by an inbound **$$ADD** record's **CODE** parameter.

Option	Description
ASCII	Data from the remote site is ASCII text strings.
EBCDIC	Data from the remote site is EBCDIC.
BINARY	Data from the remote site is binary or of unknown format.

This parameter can be overridden by the ACD file. This parameter will affect files added through WebDAV.

DIRFORM=MVS|UNIX|CLIENT|BROWSER

(Async, Bisync, FTP, SSHFTP, BP) specifies the format of the **$$DIR** command. The output of the **$$DIR** command will be compatible with the specified Connect:Enterprise or Connect:Enterprise OS/390 platform, clients, or an off-the-shelf Web browser. The default is **UNIX**. For SSHFTP, **MVS** and **CLIENT** are not valid, and **BROWSER** is the default. This parameter will affect files added through WebDAV.

DISTINGUISHEDNAME="string"

(All) Specifies the LDAP distinguished name (DN) associated with he RSD that is recolonized by the external authentication server. Use any full distinguished name. The EXTERNALAUTHRESOURSE must be set to an access definition that uses a Principal Resolver of PrincipalInRequest for the DISTINGUISHEDNAME to be used when authenticating the account.

DISCINTV=*nnnn*|NO

(Bisync) specifies the number of seconds that Connect:Enterprise waits without session activity before the session with the remote site is terminated. When **NO** is specified, it indicates that Connect:Enterprise is to wait the maximum line timeout period for the protocol/remote site type being communicated with before disconnection occurs. This parameter can be overridden by the **DISCINTV** parameter in the ACD file. The default is **NO**.

Note: When non-switched Bisync leased lines (as defined in the Bisync CPD file) are used, the default **DISCINTV=NO** should not be specified. Specifying a value for the number of seconds is necessary to return the port to a state of Connecting when the session ends.

EXTERNALAUTH=YES|NO|NEVER

Specifies if external authentication is used for this account. This field is only valid if the EXTERNALAUTH=RSD in the System Configuration.

Default = Do not use external authentication for this account. This is the default.

Yes = External authentication is used for this account.

Never = External authentication is never used for this account, even if EXTERNALAUTH=Yes in the System Configuration.

EXTERNALAUTHRESOURCE="string"

Specifies the LDAP server access definition or access definition group to be used to authenticate this account. Use any valid access definition or group.

FTPPORT=*nnnnn*

(FTP, SSHFTP) specifies the port number defined at the remote site for FTP operations. The default is **21**. See the *Connect:Enterprise Help* for information regarding controlling access to a Connect:Enterprise system running behind a packet-filtering firewall by setting port range limits in FTP operations.

FTPSCRIPT="*ipaddress,portno,loginid,password*", "*ipaddress,portno,loginid,password*",...

(FTP autoconnects) allows Connect:Enterprise FTP clients to traverse intermediate firewalls (proxy servers). The **SCRIPT** parameter and the **FTPSCRIPT** parameter are mutually exclusive; a system can use one or the other of the parameters but not both.

Two pieces of information are required for each firewall:

- IP address of the firewall host running the ftpd proxy server
- Port number that the ftpd proxy server monitors

Two pieces of information are optional:

- Valid login ID of a user of the firewall host
- Valid password for the above login ID

Note: A login ID field cannot be used without a password and the password field cannot be used without a login ID.

The format of the value coded for FTPSCRIPT is:

```
FTPSCRIPT=
"ipaddress of first firewall, portno, loginid, password;
 ipaddress of second firewall, portno, loginid, password;
    .
    .
    .
 ipaddress of nth firewall, port no, login id, password;"
```

Always code the IP addresses in dotted decimal format instead of the symbolic host names. Otherwise, an intermediate firewall may not be able to resolve the symbolic host name to its ip address.

The administrator has the option to store the ***password*** parameter in either an encrypted or clear text format. In order to encrypt any passwords, a global key must first be created and the password encryption option must be activated using the **ceukey** command.

If the password encryption option is activated, Connect:Enterprise encrypt all RSD passwords automatically, including the FTPSCRIPT firewall passwords. To encrypt the passwords in manually created or edited RSD files, the user must run the **ceupassencrypt** utility after the RSD file has been saved. See the *Connect:Enterprise Installation and Administration Guide* for more information on the **ceukey** and **ceupassencrypt** commands.

Encrypted passwords have the following format:

```
FTPSCRIPT="ipaddress, portno, loginid, ENCRYPTED_xxxxxxxx"
```

Caution: Do not edit an encrypted password.

The maximum number of characters for the **FTPSCRIPT** string is 512 characters. Users can traverse up to 9 firewalls.

FTP_PUT_OPTIONS="*string*"

(FTP, SSHFTP) specifies the FTP put options. When remotes use the FTP **put** and **mput** commands with standard syntax (for example, ftp> **put** myfile), then the Connect:Enterprise server uses **FTP_PUT_OPTIONS** to know what kind of characteristics this batch will have. For example: Is it requestable? Is it multiple requestable? Is it an ASCII batch? It then sends this information to the mailbox daemon. If **FTP_PUT_OPTIONS** is specified in both the ACD file and the RSD file, the value in the ACD file takes precedence for auto connects.

The default value for **FTP_PUT_OPTIONS** is **“”**, which means that no options are in effect.

For example, if you have set **FTP_PUT_OPTIONS= XMIT=Y** and you enter the following command:

```
ftp> put/temp/file.txt
```

Connect:Enterprise will create a batch whose batch ID is file.txt with **XMIT** effective (the requestable (R) flag will be on).

The following are the optional parameters that are allowed for the **FTP_PUT_OPTIONS** string.

Parameter	Associated Value
Mailbox ID	
Batch ID	
CODE=	A|E|B

Parameter	Associated Value
EO=	YES|NO
MULTXMIT=	YES|NO
PASSWORD=	*XXXXXXXX*
TO=	YES|NO
TRIGGER=	YES|NO
XMIT=	YES|NO

Optional Parameters for FTP_PUT_OPTIONS

The following parameters, listed alphabetically, are optional parameters when using the **FTP_PUT_OPTIONS**. In all cases, **YES** and **NO** can be abbreviated to **Y** and **N** respectively. With the exception of PASSWORD, these parameters will affect files added through WebDAV.

- CODE=A|E|B

  identifies the formats of the data being added. Three values are possible: **A** = ASCII, **E** = EBCDIC, and **B** = binary. The default is **A**.

- EO=YES|NO

  The **EO=Y** parameter specifies that this file can only be extracted once and never transmitted. If **YES** is entered, the batch is marked with a nontransmittable flag. Once extracted by the host site, it is also flagged unextractable. The default is **NO**.

- MULTXMIT=YES|NO

  enables or disables multiple transmissions of this batch. This parameter can be abbreviated to **MX**.

  YES—Indicates that the batch can be transmitted multiple times. If **YES** is specified, it overrides the **XMIT** parameter and sets it to yes. With **MULTXMIT=Y**, the added batch is flagged as multi-transmittable at the time it is added to the mailbox ID, but unlike **XMIT=Y**, the batch is not flagged as transmitted when successfully transmitted. This leaves it eligible for subsequent transmissions that would not be possible if the transmitted flag were set.

  NO—Indicates that the batch cannot be transmitted multiple times. This is the default.

- PASSWORD=*xxxxxxxx*

  enables a remote site to supply a password to the Connect:Enterprise Remote Command Exit for those mailbox sites that have enabled custom security.

  Note: The PASSWORD parameter is not available from the user interface. It can only be specified manually.

- TO=YES|NO

  enables or disables a transmit once capability.

  YES—Specifies that the batch can only be transmitted once. After transmission to the intended remote site, the batch is permanently locked and is flagged as nontransmittable and unextractable. If transmission of a batch with this parameter fails after one or more records have been transmitted, the batch is still locked. To retry the transmission, a new batch must be added from the original source.

  NO—Specifies that the batch is not flagged as unextractable. This is the default.

- TRIGGER=YES|NO

  allows files to be re-routed immediately to other remotes. In order for automatic routing to function, an auto connect file must be defined with the **CONTACT=DATA_IMMEDIATE** parameter. When the characteristics of the batch being added match the selection criteria in this ACD file, the batch will automatically be forwarded to the destination specified in the ACD file.

  YES—The batch will be re-routed if a valid auto connect list with matching selection criteria has been defined.

  NO—The batch is not forwarded. This is the default.

- XMIT=YES|NO

  determines whether a batch is limited to host site use (in the network where Connect:Enterprise exists) or can be distributed to other locations.

  YES—Specifies that the batch is available for transmission to any remote site which knows the proper mailbox ID. The batch is marked with a requestable flag. With **XMIT=Y**, the added batch is flagged as transmitted once it is successfully forwarded to another remote.

  NO—Specifies that the batch is available only for host site extraction. This indicates that the requestable flag is not set, restricting remote sites from requesting the batch. This is the default.

GETCOMMAND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent in place of the generic **MGET** command.

LOGBATCH=YES|NO

(All) enables the creation of a **LOGBATCH** receipt for the remote site. The **LOGBATCH** receipt contains the ID, BID, total bytes, batch number and the date, time and protocol that the batch was added or extracted with.

Information is added to the log batch for every add or extract operation a remote performs; the data in the **LOGBATCH** accumulates until the batch is explicitly deleted. The default is **NO**. This parameter will affect files added through WebDAV.

LOGONMSG="*string*"

(Bisync) specifies the 80-character */SIGNON card string that is used to connect to a JES2 remote type. If the Remote type is JES2 this message must be supplied or auto connect attempts to the remote site fail.

LOGOFFMSG="*string*"

(Bisync) specifies the 80-character */SIGNOFF card string that is used to sign off from a JES2 remote type.

MACS="*MAC1,MAC2,. . .*"

(SSHFTP) Specifies the message authentication code (MAC) algorithms in order of preference. The MAC algorithm is used for data integrity protection. Separate multiple algorithms with commas. The default is:

```
hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96
```

MAILBOX_LIST= "*mailbox_ID,...*"| **"*"**

(FTP and SSHFTP remote connects) indicates the mailbox IDs the user has access to. This parameter allows an administrator to provide specific users access to specific mailbox IDs, and acts as an additional layer of security. This keyword is particularly useful for limiting mailbox ID access to users of standard FTP client products, such as WS_FTP.

This parameter takes effect regardless of whether standard FTP syntax or $$ FTP syntax is used. For example, if **MAILBOX_LIST="george,steve,chris"** were specified in the RSD file for remote user george, then george would only have access to his own mailbox ID and to mailbox IDs steve and chris, regardless of whether george uses standard FTP syntax or $$ FTP syntax. The following rules apply:

- A value of **"*"** indicates access to all other mailbox IDs. This is the default.

- A value of **""** (empty string) restricts the user to their own mailbox ID only.

- Values should always be surrounded by double quotes.

- The comma-separated list can be up to 256 characters.

- Wildcard specifications are not allowed.

- The user does not need to specify their own remote login ID in the **MAILBOX_LIST** specifications; it is implied (Connect:Enterprise will add the login ID automatically). This parameter will affect files added through WebDAV.

MBXSEP=YES|NO

(FTP, SSHFTP) dictates the behavior of the Connect:Enterprise FTP client for auto connects. It specifies how batches will be separated if they are coming from different mailboxes. This parameter works with the **BCHSEP** parameter. This parameter will affect files added through WebDAV.

MODE=SENDRECEIVE|SENDONLY|RECEIVESEND|RECEIVEONLY

(Async, Bisync, FTP, SSHFTP, BP) specifies the sequence in which communications with the remote site occurs during an auto connect. This parameter can be overridden by the **MODE** parameter in the ACD file.

Option	Description
SENDRECEIVE	Connect:Enterprise first sends batches to the remote, then it turns the line around to receive batches from the remote. This is the default.
SENDONLY	Connect:Enterprise sends batches to the remote, then it disconnects from the remote site.
RECEIVESEND	Connect:Enterprise first receives batches from the remote site, then it turns the line around to send batches to the remote.
RECEIVEONLY	Connect:Enterprise receives batches from the remote site, then it disconnects from the remote site.

PASSWORDAUTH=YES|NO

(SSHFTP) Specifies whether the remote client can perform password authentication. This parameter is only valid if you are connecting to a SSHFTP server using a schedule (auto connect).

Yes = The remote client can perform password authentication.

No = The remote client cannot perform password authentication.

PASSWORD="*password*"

(All) identifies the 64 character password that the remote site needs to supply during a remote connect. This parameter is also used to specify passwords in RSD for local users.

If a **PASSWORD** parameter is entered here and the remote site does not supply a password, or supplies one that differs from the one specified in the RSD, the connection is terminated. This string value is case sensitive.

If you want to allow a remote site to omit the use of the **PASSWORD** parameter on inbound $$ cards, the RSD must include a **PASSWORD** parameter that specifies a value equal to the mailbox ID that the remote specifies on the $$ card.

If **PASSWORD** is not specified or is null in a remote's RSD file, the remote does not have to send a password. All password values the remote sends are regarded as valid.

The **PASSWORD** parameter is required with Connect:Enterprise (Secure FTP). The administrator has the option to store RSD passwords in either an encrypted or clear text format. In order to encrypt RSD passwords, a global key must first be created and the password encryption option must be activated using the **ceukey** command. This parameter will affect files added through WebDAV.

If the password encryption option is activated, Connect:Enterprise encrypts all RSD passwords automatically. To encrypt the passwords in manually created or edited RSD files, you must run the **ceupassencrypt** utility after the RSD file has been saved. See the *Connect:Enterprise Installation and Administration Guide* for more information on the **ceukey** and **ceupassencrypt** commands.

Unencrypted passwords have the following format:

```
PASSWORD="password"
```

Encrypted passwords have the following format:

```
PASSWORD="ENCRYPTED_xxxxxxxx"
```

Caution: Do not edit an encrypted password.

PORT_RANGE=*"lllll–hhhhh"*

(FTP) specifies the pool of ports assigned for new socket operations. Use up to five numeric ranges where the format "*lllll-hhhhh*" represents the beginning and end of each range. The low element of the range denoted as 'lllll' must be a number greater than or equal to 1025. The high element of the range denoted as 'hhhhh' must be a number less than or equal to 65535. The low element must be numerically less than the high number. The string must be enclosed in quotation marks.

PORT_RETRIES=*nn*

(FTP) specifies the number of times the pool of ports is checked for an available port. The numeric value ranges from 0 to 99 with a default value of 0 (zero), or no retries.

PORT_RETRY_WAIT_TIME=*nnn*

(FTP) specifies the number of seconds to wait before the next attempt to connect to the port. The numeric value ranges from 0 to 180 with a default value of 0 (zero).

POSTRECEIVE="*string*"

(FTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately after the GET command.

POSTSEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately after the PUT command.

Enter the appropriate string in the field provided.

PREFERREDAUTH="*password;publickey;password,publickey;publickey,password*"

(SSHFTP) Specifies the authentication method the remote client prefers to use. The following methods are available:

Method	Description
password	The remote will only use password authentication.
publickey	The remote will only use public key authentication.
password,publickey	The remote prefers password authentication. If password authentication is not available on Connect:Enterprise, or if password authentication fails, the client will use public key authentication. You must set PASSWORDAUTH=YES and PUBKEYAUTH=YES. You must also create the following file and put the client's public key in the file: $CMUHOME/ssh/users/*account name*/authorized_keys.
publickey,password	The remote prefers public authentication. If public key authentication is not available on Connect:Enterprise, or if public key authentication fails, the client will use password authentication. You must set PASSWORDAUTH=YES and PUBKEYAUTH=YES. You must also create the following file and put the client's public key in the file: $CMUHOME/ssh/users/*account name*/authorized_keys.

This parameter is only valid if you are connecting to a SSHFTP server using a schedule (auto connect).

PRERECEIVE="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the GET command.

Enter the appropriate string in the field provided.

PRESEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the PUT command.

PUBKEYAUTH=YES|NO

(SSHFTP) Specifies whether to attempt public key authentication. This parameter is only valid if you are connecting to a SSHFTP server using a schedule (auto connect).

Yes = The remote client can perform public key authentication.

No = The remote client cannot perform public key authentication.

RECORDSEPARATOR=CR|CRLF|LF|1E|1F

(Async, Bisync) identifies the end-of-line (EOL) character(s) to be used during the auto connect

Option	Description
CR	Specifies that end-of-lines are delimited by a carriage return.

Option	Description
CRLF	Specifies that end-of-lines are delimited by a carriage return followed by a line feed.
LF	Specifies that end-of-lines are delimited by a line feed. This is the default for Async.
1E Bisync only	This parameter sets the record separator that will be used for transmission of nontransparent data to the remote site. This is the default for Bisync.
1F Bisync only	This parameter sets the record separator that will be used for transmission of nontransparent data to the remote site.

REMOTEFILENAME="*user_string*"

(FTP, SSHFTP Async with ZMODEM or KERMIT) specifies the destination file name on the remote system during FTP auto connects. This parameter can be no longer than 256 characters. Anything contained within the string will not be translated (for example **XMIT=Y** or anything similar). The string is simply forwarded. This string may not contain spaces; however, all printable ASCII characters are allowed. If this parameter is coded in the RSD and the ACD, the value in the ACD file will take precedence.

RENAMEFILE|RENAME_FILE |REN_FILE= YES|Y|NO|N

(FTP, SSHFTP) provides for backward compatibility for FTP users using versions of Connect:Mailbox UNIX prior to version 3.1. If **YES** is chosen, whenever FTP does a file transfer, the previous Connect:Mailbox method will be retained. For example, the file XYZ is transferred to the remote's directory in a file called XYZ.tmp and then renamed XYZ.tmp to XYZ. If **YES** is chosen, the parameter **SUNIQUE** is ignored.

If No is chosen, an FTP file transfer will occur without any renaming. Also, if **NO** is chosen, the value specified by **SUNIQUE** is valid and the remote user can determine whether or not to prevent files from being overwritten using that parameter.

RMT_FNAME_LEN|REMOTE_FILENAME_LENGTH=SHORT|LONG

(FTP, SSHFTP) specifies the format for files created on the local host so that it conforms with the format of files existing on the local host.

When **SHORT** is specified, file names created on the local system with mget must conform to the 8.3 DOS convention (for example, batch_number.dat) where batch_number is an eight-digit number like 00004075. SHORT is only valid for remote connections.

When **LONG** is specified, file names can be free-format, and are of the form bid.batch_number, where bid is the batch ID and batch_number is the batch number (without leading zeroes).

The **RMT_FNAME_LEN** keyword is only meaningful when **BCHSEP=OPT4**. Otherwise, the **RMT_FNAME_LEN** keyword is ignored.

RMT_PASSWORD="*password*"

(FTP auto connects, SSHFTP, BP) indicates the password on the remote system. If the protocol is BP, it is the password for the GIS account. This keyword affects auto connects only. The default is the password specified by the password keyword.

The administrator has the option to store the **RMT_PASSWORD** parameter in either an encrypted or clear text format. In order to encrypt any passwords, a global key must first be created and the password encryption option must be activated using the **ceukey** command.

If the password encryption option is activated, Connect:Enterprise encrypts all RSD passwords automatically, including the RMT_PASSWORD parameter. To encrypt the passwords in manually created or edited RSD files, the user must run the **ceupassencrypt** utility after the RSD file has been saved. See the *Connect:Enterprise Installation and Administration Guide* for more information on the **ceukey** and **ceupassencrypt** commands.

Encrypted passwords have the following format:

```
RMT_PASSWORD="ENCRYPTED_xxxxxxxx"
```

Caution: Do not edit an encrypted password.

RMT_USER="*username*"

(FTP auto connects, SSHFTP, BP) indicates username is the user name on the remote system. If the protocol is BP, it is the user ID for the GIS account. This parameter affects auto-connects only. The default is the specified RSD name.

SESSIONEND="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during an FTP auto connect. The string is sent immediately before the **QUIT** command.

SESSIONSTART="*string*"

(FTP, SSHFTP) specifies a string to be sent to the remote site during a FTP auto connect. The string is sent immediately after the **USER** and **PASS** commands.

SECURITY_PROTOCOL_FILE=*/path/spd*

(Secure FTP only) specifies the name of the SPD file to be used during the FTP session. For a secure session to be allowed, the FTP CPD file must also specify an SPD that has **SECURITY_POLICY=OPTIONAL** or **SECURITY_POLICY=REQUIRED**. The SPD filename must be a valid path on the host containing the FTP communication daemons. Refer to Chapter 7, *Security Protocol Definitions*, for more information regarding SPD files.

The override functions of the FTP RSD **SECURITY_PROTOCOL_FILE** parameter operate differently for auto connects and remote connects.

Auto connects—The SPD file specified in the RSD overrides the SPD file specified in the corresponding CPD. However if **SECURITY_POLICY**=REQUIRED in the SPD file specified in the CPD, then **SECURITY_POLICY** must also be REQUIRED in the SPD file

specified in the RSD. If **SECURITY_POLICY**=OPTIONAL in the SPD file specified in the CPD, then **SECURITY_POLICY** can be either OPTIONAL or REQUIRED in the SPD file specified in the RSD. To use implicit SSL for the account, specify an SPD file with **SECURITY_POLICY**=IMPLICIT. This will enable any SSL-enabled FTP server to be used by the account for implicit SSL. To use a specific SSL-enabled FTP server for the account, do not specify an SPD file. Instead, in the ACD file, specify an implicit SSL server as the auto-connect resource. For more information, see *RESOURCE=daemon name:resource name,...* on page 40.

Remote connects—The **SECURITY_POLICY** parameter of the SPD file specified in the RSD overrides the **SECURITY_POLICY** parameter of the file specified in the corresponding CPD file. All other parameters function as set in the CPD file.

SCAN=YES|NO

(Bisync) controls how Connect:Enterprise will process inbound batches that contain embedded **$$ADD** cards.

NO—Embedded **$$ADD** cards will not be processed. They will be considered data.

YES—Embedded **$$ADD** cards will cause subsequent data to be added as a separate batch. This is the default.

In the case where **SCAN=NO** is specified in the RSD file, **SCAN=YES $$ADD** card parameters will be honored only on physical batch boundaries. This is also true when a **SCAN=NO $$ADD** card parameter is specified, regardless of the setting of the SCAN RSD parameter. After a **SCAN=NO $$ADD** card parameter has been processed, no more embedded **$$ADD** cards will be processed until the next physical batch boundary. A physical batch boundary is defined as the first inbound record of a session or the first inbound record following an ETX or EOT.

Scanning logic will be reset to the value specified in the RSD file at each ETX or EOT.

Caution: When **SCAN=YES** is enabled, either through the **SCAN** RSD parameter or the **SCAN $$ADD** card parameter, embedded **$$ADD** cards will be scanned (sought). This will occur on every record in a nontransparent mode transfer. Scanning is not supported in transparent mode transfer.

SCRIPT=BEGIN...END

(Async) initiates the parsing of an optional user-supplied Async script that can facilitate conversational Async sessions. **SCRIPT** is used for Async remotes only and the only purpose is to simulate a user login to a remote site. The script starts with the keyword **BEGIN** and is terminated with **END**. The lines that fall between these delimiters use one of the following keywords: **SEND**, **EXPECT**, **AUXPARMS**, **PAUSE**, **UPLOAD**, or **DOWNLOAD**.

The following is a sample script.

```
SCRIPT=Begin
  SEND  "\r\n\r\n"
  EXPECT "login:",3,1,break
   SEND  "johnqpub\n"
  EXPECT "pass:",3,1,break
   SEND  "password\n"
  EXPECT "prompt>",3,1,break
  SEND "rz\n"
   PAUSE 1
  UPLOAD
   EXPECT "prompt>",3,1,break
  SEND  "cd download; sz *\n"
   PAUSE 1
   DOWNLOAD
  EXPECT "prompt>",3,1,break
  SEND  "exit\n"
End
```

- SEND “string”

  specifies a string, enclosed in double quotes, to be transmitted to the remote site. This string can contain standard C notation such as \r (carriage return) or \n (newline) in addition to text.

- EXPECT “string”,retries,seconds[,“response”|,break]

  specifies a string, enclosed in double quotes, that is anticipated as received data from the remote site. It must also specify the number of seconds to wait for the inbound string and the number of times to repeat the wait-and-test period. If the test is satisfied before the specified number of retries expire, the next instruction in the script will be executed. Each time a wait of the specified duration expires without receipt of the expected inbound data, the response string will be transmitted to the remote site and the number of retries will be decreased. If the **EXPECT** fails completely, the response string will have been transmitted on the last retry and then the script is aborted. The response argument is optional and defaults to a *\r* (a carriage return is transmitted). In lieu of a string enclosed in double quotes, the value *break* may be specified as a response. This causes a standard Async 250ms break to be transmitted.

  The following sample shows using **SCRIPT** to control login to an Async Remote site during an auto connect session:

```
SCRIPT=Begin
        SEND “\r\n\r\n”
        EXPECT “login:”,3,1,break
        SEND “jsmith\r”
        EXPECT “password:”,9,2
        SEND "mypasswd\r"
    End
```

  This example script will send two carriage return-new line pairs, then it will await the string *login:* and respond with *jsmith* and a carriage return. It will similarly await the string *password* and respond with *mypasswd* and a carriage return. With the execution of the first

EXPECT statement, after three retries, waiting for 1 second each, and transmitting a break after each wait period, the expected string has not been collected, the script will abort. Execution of the second EXPECT statement would also abort the script if, after 9 retries, waiting 2 seconds each, and transmitting a carriage return (the default response) after each wait period, the expected string *password:* had not been received.

- AUXPARMS=8|7,E|EVEN|O|ODD|N|NONE,1|2

  allows for a logging script during an auto connect to use alternative number of bits, parity and stop bits for the serial ports. This parameter supports command interaction with the alternative settings. However, after the Async auto connect process has logged into the remote host, all file transfers occur in 8 bit, no parity and 1 stop bit mode. The default is **8, N, 1**.

- PAUSE *seconds*

  specifies a number of seconds for the script to pause before continuing to the next line of code.

- UPLOAD

  starts the send process in the script. The protocol specified in the RSD parameter **PROTOCOL** is used. The script continues upon completion of the upload.

- DOWNLOAD

  starts the receive process in the script. The protocol specified in the RSD parameter **PROTOCOL** is used. The script continues upon completion of the download.

SENDBUFF=*nnnn*

(Bisync) determines the maximum length of the block (in bytes) for transmission to this particular remote site. The maximum size is 4096 bytes. The default it **4096** bytes. If it is set to 0 or to a value greater than 4096, the default value will be used.

For Cleo Bisync SYNCcable+ connections the default value, 4096, is valid for both transparent and non-transparent data.

For ARTIC Bisync connections, the valid values for **SENDBUFF** depend on the transparency of the data you will be transmitting. If you plan to transmit only non-transparent (EBCDIC) data to and from a remote, a **SENDBUFF** value greater than 2048 is acceptable in the RSD for that remote. However, if you plan to also transmit transparent (ASCII or binary) data to the same remote, **SENDBUFF** must be limited to 2048 or less. This may increase the wall-time required for EBCDIC transmissions, but eliminates the need to maintain separate RSD files for transparent and non-transparent transmissions to and from the same remote.

Consider if the following conditions all apply:

- you plan to transmit both transparent and non-transparent data to the remote
- speed is critical for EBCDIC transmissions

If the above conditions apply, using two separate RSD files: one for transmission of EBCDIC batches and one for transparent batches might be beneficial. The RSD for EBCDIC

transmissions should specify **SENDBUFF=4096**, and the RSD for transparent batches should specify **SENDBUFF=2048** or less.

SFI=YES|NO

(Bisync) suppresses the final IRS in each Bisync message block, producing a transmission with the message block formatted as in the two examples in the following figure:

```
02 - record - 26
02 - record - 1E - record - 1E - record - 26
```

If **SFI=NO**, the parameter constructs outbound Bisync message blocks with an IRS following each record in the block. This is the default. For example, an unblocked transmission (1 record per block) has message blocks formatted as in the following figure:

```
02 - record - 1E - 26
```

Blocked transmissions (more than one logical record per block) have message blocks formatted as in the following figure, where the block contains three records:

```
02 - record - 1E - record - 1E - record - 1E - 26
```

TRANSLATE="*filename*"

identifies the name of the file that contains the table for ASCII-to-EBCDIC and EBCDIC-to-ASCII translation. The specified string should not include path information. The translate table is assumed to be in the *$CMUHOME*/xlate directory. The default file name is *ascebd.tbl.* For more information, refer to the *Translate Table Format* appendix in the *Connect:Enterprise Installation and Administration Guide.*

TRUNC=YES|NO

(Bisync) identifies whether trailing blanks should be truncated before transmission to the remote site. The default is **NO**.

TRUSTHOSTKEY=YES|NO

(SSHFTP) If an SSHFTP server sends a host key, and that host key does not have a match in the $CMUHOME/users/*accountname*/known_hosts file, this parameter controls whether to trust the host key sent. If TRUSTHOSTKEY=YES, the host key is stored in the known_hosts file, and the connection is accepted. If TRUSTHOSTKEY= NO, the auto connect is rejected. If an auto connect is rejected for this reason, the only way the auto connect will be successful is to change the parameter to Yes or manually add the server's host key to the known_hosts file.

This parameter is only valid if you are connecting to a SSHFTP server using a schedule (auto connect).

TYPE=REMOTE|CMUMBOX|JES2

(FTP, Bisync) indicates if an enhanced protocol (CMUMBOX) is used to communicate with the remote site.

Option	Description
REMOTE	The remote site is a typical remote site that communicates with Connect:Enterprise using standard protocols and command formats. This is the default.
CMUMBOX	The remote site is another Connect:Enterprise site. This is an enhanced protocol.
JES2 (Bisync only)	The remote site is a JES2 site.

Sample RSD File

Each of the following remotes have been coded with every RSD parameter applicable for its protocol. Every keyword specifies a default value if a default exists. Some parameters do not have defaults, so fictitious values have been supplied. All RSD parameters are case-insensitive, but for clarity here, all required keywords are upper case. All default values are upper case.

The following is a sample Async remote site RSD (interactive mode).

```
PHONE = "5551212","5552129"
PROTOCOL = zmodem
AUTOCONVERT = NONE
BCHSEP = NONE
CONCATFILES = NO
DATAFORMAT = ASCII
DIRFORM = UNIX
DISCINTV = NO
LOGBATCH = yes
MODE = SENDRECEIVE
PASSWORD = "password"
RECORDSEPARATOR = LF
SCRIPT = BEGIN
      AUXPARMS = 8,N,1
      SEND "\r\n\r\n"
      EXPECT "login:",3,1,break
      SEND "jsmith\r"
      EXPECT "password:",9,2
      SEND "mypasswd\r"
     End
TRANSLATE = "ascebd.tbl"
```

The following is a sample Async remote site RSD (non-interactive mode).

```
PHONE = "5551212"
PROTOCOL = xmodem
AUTOCONVERT = NONE
BCHSEP = NONE
CONCATFILES = NO
DATAFORMAT = ASCII
DIRFORM = UNIX
DISCINTV = NO
LOGBATCH = yes
MODE = SENDRECEIVE
PASSWORD = "password"
RECORDSEPARATOR = LF
TRANSLATE = "ascebd.tbl"
```

The following is a sample FTP remote site RSD.

```
ADDRESS = "999.999.999.999","111.111.111.111"
FTPPORT = 21
PROTOCOL = ftp
ACRECVDIR = /home/ftp/download
ACSENDDIR = /home/ftp/upload
AUTOCONVERT = NONE
BCHSEP = NONE
DATAFORMAT = ASCII
DIRFORM = UNIX
DISCINTV = NO
FTP_PUT_OPTIONS = "string"
FTPSCRIPT = "ipaddress, portno.loginid,password",
 "ipaddress, portno.loginid,password",...
LOGBATCH = yes
PORT_RANGE="10024-10029, 30024-30099"
PORT_RETRIES=0
PORT_RETRY_WAIT_TIME=30
MAILBOX_LIST = "*"
MODE = SENDRECEIVE
PASSWORD = "password"
REMOTEFILENAME = "user_string"
RMT_PASSWORD = "password"
RMT_USER = "username"
SCAN = YES
TRANSLATE = "ascebd.tbl"
TYPE = REMOTE
```

The following is a sample SSHFTP remote site RSD.

```
ADDRESS = "999.999.999.999","111.111.111.111"
FTPPORT = 22
PROTOCOL = ftp
ACRECVDIR = /home/ftp/download
ACSENDDIR = /home/ftp/upload
AUTOCONVERT = NONE
BCHSEP = NONE
COMPRESSION = yes
CIPHERS =
"aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc"
CHECKHOSTIP = YES
DATAFORMAT = ASCII
DIRFORM = UNIX
DISCINTV = NO
FTP_PUT_OPTIONS = "string"
LOGBATCH = yes
MACS = "hmac-md5,hmac-sha1,hmac-ripemd160,hmac-sha1-96,hmac-md5-96"
KNOWN_HOSTS = /ssl/known_hosts
PASSWORDAUTH = Yes
PORT_RANGE="10024-10029, 30024-30099"
PORT_RETRIES=0
PORT_RETRY_WAIT_TIME=30
PREFERREDAUTH="publickey,password"
PROTOCOL= sshftp
PUBKEYAUTH = YES
MAILBOX_LIST = "*"
MODE = SENDRECEIVE
PASSWORD = "password"
REMOTEFILENAME = "user_string"
RMT_PASSWORD = "password"
RMT_USER = "username"
TRANSLATE = "ascebd.tbl"
TRUSTHOSTKEY = YES
TYPE = REMOTE
```

The following is a sample Bisync remote site RSD.

```
PHONE = "5551213"
PROTOCOL = bsc
AUTOCONVERT = NONE
BCHSEP = NONE
BLOCK = 5
COMPRESSION = NO
CONCATETX = NO
DATAFORMAT = EBCDIC
DIRFORM = UNIX
DISCINTV = NO
LOGBATCH = yes
MODE = SENDRECEIVE
PASSWORD = "password"
SCAN = YES
SFI = NO
TRANSLATE = "ascebd.tbl"
TYPE = REMOTE
```

The following is a sample RSD that defines the password for a local site user.

```
LOGBATCH = YES
PASSWORD = "password"
TRANSLATE = "ascebd.tbl"
```

Configuring RSD Files for Alternate Routing

If your remote site supports multiple protocols, the alternate routing feature provides a mechanism of transporting data through a secondary method if your primary method fails during send operations through auto connects. Alternate routing is used only for **MODE=SENDONLY** auto connects.

Alternate routing uses the usual RSD files that define each connection but are named in a particular convention that signifies the order of preference. If the primary connection profile (RSD file) is named ABC, then the secondary connection profile would be named ABC@1.

The format for the alternate profiles is *remote_name@n*, where *n* is a number greater than or equal to 1. Remote ABC would have one RSD files named ABC. The alternate profile can be stored in another RSD file with the name ABC@1. The number *n* indicates the order in which the RSDs will be used. There could be another profile in the file ABC@2 if your remote site has the capabilities for three communication protocols. The profile stored in ABC is always the first one used, ABC@1 will be the second, ABC@2 the third, and so forth. There is no limit to the number of profiles that may be set up. For example, remote site ABC has both FTP and Async capabilities but it prefers using FTP for receiving data.

The RSD file ABC could be defined as:

```
ADDRESS="ftp.abc.com"
FTPPORT=10078
PROTOCOL=FTP
Password="letmein"
Bchsep=OPT3
```

The RSD file ABC@1 could be defined as:

```
ADDRESS="my_node"
PROTOCOL=ZMODEM
Bchsep=OPT4
```

Sample

The following scenario is an example of how alternate routing performs for the following ACD file:

```
Sessions=1
Interval=1
Requeues=1

Remotes="ABC"
    Mode=SendOnly
    Sendid="payroll"
```

1. An auto connect process starts (manually, timer-driven, or data driven).
2. The auto connect process asks the Control daemon for the RSD information for the remote ABC.
3. The Control daemon parses the RSD file for the remote ABC. The protocol defined in the file ABC is FTP.
4. The FTP process, for whatever reason, fails to send the batches for mailbox ID=payroll to the address ftp.abc.com.
5. The FTP process indicates to the auto connect process that an error has occurred.
6. Since requeues is set to 1 in the ACD file, the auto connect process attempts to start the auto connect again after having waited for an interval of 1 minute (since Interval is also defined as 1).
7. This process repeats until either the FTP process indicates that the transfers succeeded or the requeues count is exhausted.
8. When the auto connect has failed to send data to the remote ABC through the preferred protocol FTP, it will ask the Control daemon for the next preferred profile.
9. The Control daemon reads the RSD directory to see if there are any ABC@* profiles. If it finds such a profile, it will select the profile with the least *n* that is not equal to the last profile tried by the auto connect.
10. The auto connect tries each of the fallback profiles the number of times specified by the **REQUEUES** parameter.
11. If it does not find any more profiles the auto connect will fail.

Guidelines for Alternate Routing

The **RESOURCE** parameter will not be used by the auto connect if the alternate routing profile is used. Ordinarily, the parameters defined in the ACD file override the respective values defined the RSD files.

However, when an alternate RSD file is being used, protocol-specific parameters such as **ADDRESS**, **PHONE**, **FTPPORT**, **BLOCK**, **SFI**, **ACRECVDIR**, **ACSENDIR**, **CONCATETX**, **CONCATFILES**, **SCAN**, **COMPRESSION**, and **BCHSEP** as defined in the alternate RSD file take precedence over the respective values defined in the ACD. Adhere to the following guidelines when setting up RSD files for alternate routing.

1. Do not code **RESOURCE=** in the ACD file.
2. Supply **PROTOCOL=** in all the versions of the RSD files.
3. Supply **ADDRESS** or **PHONE** in all the versions of the RSD files.
4. Code all protocol-specific parameters in the RSD files. Supply even the default values.
5. Supply **PASSWORD** only in the main RSD profile.

Chapter 7

Security Protocol Definitions

The Security Protocol Definitions (SPD) files contain parameters that address secure communications and are valid only with Connect:Enterprise (Secure FTP). At least one SPD file is required to enable Secure FTP, but more can be created. The appropriate SPD file may be specified in ACD, RSD, and CPD files with the **SECURITY_PROTOCOL_FILE** parameter. The SPD files must reside on the same host as the FTP communications daemons.

The information for the SPD file(s) is a direct result of the steps outlined in Appendix C, *Secure FTP*, in the *Connect:Enterprise Installation and Administration Guide*. Before creating or modifying an SPD file, refer to the *Connect:Enterprise UNIX Installation and Administration Guide* on your documentation CD.

SPD File Conventions

When you create an SPD file, adhere to the following conventions:

- Specify SPD parameters in any sequence.
- Place each parameter on a single line.
  - Do not span multiple lines with a single parameter.
  - White space lines between parameters will be ignored.
- Parameters and keywords are not case-sensitive.

SPD File Format

Required parameters are bold in the following table. An overview of each parameter follows the table.

Parameter	Associated Values
SECURITY_POLICY=	REQUIRED|DISALLOWEDOPTIONAL|IMPLICIT
ROOT_CERT_FILE=	*filename*
RC_KEYCERT_FILE=	*filename*
CIPHER_STRENGTH=	STRONG|EXPORT|ALL
CIPHER_SUITES=	"*cipher suite:cipher suite:...*"
AUTH=	ServerOnly|ServerClient
SSL_CLIENT_CCC_POLICY=	REQUIRED|OPTIONAL|DISALLOWED
SSL_SERVER_CCC_POLICY=	REQUIRED|OPTIONAL|DISALLOWED
VERIFYSERVERCOMMONNAME=	YES|NO

Required Parameters

The following parameter is required.

SECURITY_POLICY=REQUIRED|OPTIONAL|DISALLOWED|IMPLICit

specifies the server security policy. This parameter is required.

If **DISALLOWED** is specified at the CPD level, security is completely turned off on the server side; for example, it cannot be enabled in RSD files. On the client side, Secure FTP negotiation will not be attempted unless it is activated at the RSD or ACD level.

If **REQUIRED** is specified, Connect:Enterprise UNIX attempts Secure FTP negotiation with the remote and ends the session if the remote does not support Secure FTP.

If **OPTIONAL** is specified, Connect:Enterprise UNIX attempts Secure FTP negotiation with the remote, but continues with a nonsecure session if the remote does not support Secure FTP.

If **IMPLICIT** is specified, a value of "required" is applied for all incoming FTP transfers. The FTP server instance will expect the SSL negotiation to begin immeditaely after the TCP/IP socket connection for the command socket. If the FTP client fails to immediately begin SSL negotiation, the connection will be closed with no feedback to the client.

Note: If an SPD associated with the account (RSD) and the security policy in that SPD differs from the security policy setting in the SPD attached to the auto-connect resource, the SPD associated with the account takes precedence.

Optional Parameters

The following parameters, listed alphabetically, are optional.

AUTH=ServerOnly|ServerClient

specifies the authentication type used for connections. Default is **ServerOnly**.

If **AUTH=ServerOnly** is specified, server-only authentication occurs and the client establishes the connection after verifying the server certificate during the initial handshake.

If **AUTH=ServerClient** is specified, client-server authentication occurs and the server establishes the connection after first authenticating itself to the client and then requesting the client certificate and verifying it during the initial handshake. Both a trusted root certificate file (trusted.txt) and a key/cert file (keycert.txt) are necessary for this bi-directional authentication. If either the trusted root certificate file or key/cert file can not be located or is invalid, the handshake fails.

CIPHER_STRENGTH=STRONG|EXPORT|ALL

specifies the types of cipher suites that are allowed.

Option	Description
STRONG	Allows only suites with greater than 40-bit encryption.
EXPORT	Allows only suites with 40-bit encryption.
ALL	Allows all suites supported by Connect:Enterprise UNIX.

Refer to the *Connect:Enterprise UNIX Installation and Administration Guide* for important information regarding cipher suites.

CIPHER_SUITES=*"cipher suite:cipher suite:..."*

specifies the cipher suites, in order of preference, allowed by Connect:Enterprise UNIX. A cipher suite consists of a key exchange method, a data encryption method, and a message digest function (MD5 or SHA).

Supported cipher suites are:

- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_NULL_MD5

- SSL_RSA_WITH_NULL_SHA

ROOT_CERT_FILE=*filename*

specifies the file containing the signing certificates of approved certificate authorities. A trusted root, *$CMUHOME/spd/trusted.txt*, is provided as part of the Connect:Enterprise UNIX installation. Edit this file, or refer to a different file, to add or remove certificate authorities. Certificates must be in X.509, BER-encoded, base64-encoded PEM format. Your certificate authority can provide technical details on your certificate.

RC_KEYCERT_FILE=*filename*

specifies the file containing the Connect:Enterprise UNIX server certificate chain and private key. The private key must be in PKCS#8, BER-encoded, base64-encoded format. The certificate must be in X.509, BER-encoded, base64-encoded format. Your certificate authority can provide technical details on your certificate.

SSL_CLIENT_CCC_POLICY=*REQUIRED*|*OPTIONAL*|***DISALLOWED***

Specifies whether FTP auto connection attempts to send the CCC command. If the command is accepted, the control connection operates in clear text for the remainder of the session. Each endpoint of the connection must support the use of the CCC command.

If **REQUIRED** is specified, CCC is required and must be accepted for the session.

If **OPTIONAL** is specified, CCC is attempted and, if accepted, it is honored. If it is rejected, the session remains active and the control connection remains encrypted.

If **DISALLOWED** is specified, CCC is not attempted by the FTP client. This is the default value.

SSL_SERVER_CCC_POLICY=*REQUIRED*|*OPTIONAL*|***DISALLOWED***

Specifies whether the FTP server accepts the CCC command if it is sent by the client. Each endpoint of the connection must support the use of the CCC command.

If **REQUIRED** is specified, the SSL FTP server must process the CCC before any data port operation can be attempted.

If **OPTIONAL** is specified, the CCC command is honored if it is sent by the client. No error results if the client does not send the CCC command.

If **DISALLOWED** is specified, the CCC command is not honored. The session remains active and the control connection remains encrypted. This is the default value.

VERIFYSERVERCOMMONNAME=YES|NO

Specifies whether common name validation is performed when authenticating an SSL certificate.

- Yes—If common name information is in the certificate, Connect:Enterprise performs the common name validation.
- No—Connect:Enterprise UNIX does not perform common name validation.

Sample SPD File

The following figure shows the basic SPD file format:

```
ROOT_CERT_FILE=/usr/tcom/cmunix/spd/trusted.txt
RC_KEYCERT_FILE=/usr/tcom/cmunix/spd/keycert.txt
SECURITY_POLICY=OPTIONAL
CIPHER_STRENGTH=ALL
CIPHER_SUITES="SSL_RSA_WITH_RC4_128_MD5:SSL_RSA_WITH_RC4_128_SHA:SSL_RSA_WITH_3DES_E
DE_CBC_SHA"
AUTH=ServerOnly
```

Chapter 8

Authentication Server Configuration

The Authentication Server configuration file (passadmpf.cfg) is used to set the password policy defaults. Before modifying the passadmpf.cfg file, refer to the Password Administration chapter of the *Connect:Enterprise UNIX Installation and Administration Guide*.

ASC File Conventions

When you create an ASC file, adhere to the following conventions:

- Specify ASC parameters in any sequence.
- Place each parameter on a single line.
  - Do not span multiple lines with a single parameter.
  - White space lines between parameters will be ignored.
- Enter comments preceded by the # character. When the # character is found, the rest of the line is considered to be a comment.

Note: Parameters and keywords are not case-sensitive.

ASC File Format

All of the parameters in the following table are required. A sample ASC file follows the table. All parameters except EMAIL and POLICY_LEVEL can be overridden when you create a password policy.

Parameter	Associated Value
POLICY_LEVEL	0 = (default) Disallowed; RSD policy files are not evaluated. 1 = Optional; if the RSD policy file exists for the user, it is evaluated. 2 = Required; all users who log on to Connect:Enterprise UNIX must have an RSD policy file.
DURATION_FLAG	Y = Users must change their passwords after the number of days indicated in DURATION. N = (default) No DURATION is enforced.
CONSECUTIVE_FAILED_LOGON_FLAG	*Y = CONSECUTIVE_FAILED_LOGON_ATTEMPTS limit is enforced.* N = (default) *CONSECUTIVE_FAILED_LOGON_ATTEMPTS limit is not enforced.*
CONSECUTIVE_FAILED_LOGON_ATTEMPTS	Number of failed logon attempts before the account is locked. Range is 1–999. The default is 5. You must set CONSECUTIVE_FAILED_LOGON_FLAG to Y for this parameter to be enforced.
DURATION	Number of days a password is valid before it must be changed. You must set DURATION_FLAG to Y for this limit to be enforced. Range is 1–366. Default is **366**.
EMAIL	Specifies the e-mail address of the person to notify when an account is locked because of consecutive failed logon attempts in the following format: server:port/support@ceunix.customer.com
PASSWORD_HISTORY_FLAG	*Y* = Prevents users from changing their password to a password that they have used before. The system checks their password history and fails if a new password matches an old password. The number of passwords saved in their history is defined in the PASSWORD_HISTORY parameter. N = (default) PASSWORD_HISTORY is not enforced.
PASSWORD_HISTORY	Indicates the number of passwords that are saved in the system. Users cannot change their passwords to any password saved in the system. Range is 1–99. Default is **99**.
PASSWORD_LENGTH_FLAG	*Y* = the MAXIMUM_PASSWORD_LENGTH and MINIMUM_PASSWORD_LENGTH fields are enforced. N = (default) the MAXIMUM_PASSWORD_LENGTH and MINIMUM_PASSWORD_LENGTH fields are not enforced.
MAXIMUM_PASSWORD_LENGTH	Indicates the maximum allowed length of a password. Range is 1–64. Default is **64**.
MINIMUM_PASSWORD_LENGTH	Indicates the minimum allowed length of a password. Range is 1–64. Default is **1**.

Sample ASC File

The following figure shows the basic ASC file format.

```
#Authentication Server Defaults
EMAIL=server01:80/support@ceunix.customer.com
POLICY_LEVEL=1
DURATION_FLAG=N
DURATION=120
CONSECUTIVE_FAILED_LOGON_FLAG=N
CONSECUTIVE_FAILED_LOGON_ATTEMPTS=5
PASSWORD_HISTORY_FLAG=N
PASSWORD_HISTORY=6
PASSWORD_LENGTH=N
MINIMUM_PASSWORD_LENGTH=6
MAXIMUM_PASSWORD_LENGTH=64
```

Appendix A

Site Administration User Interface Terminology

This chapter maps the Connect:Enterprise Site Administration user interface terminology with Connect:Enterprise UNIX configuration files and utilities. Use these tables to familiarize yourself with the terminology that the Site Administration interface presents.

Mapping the Site Administration Interface to the Configuration Files

The following table identifies the Site Administration user interface tasks and the associated configuration files:

Navigation Bar Heading	User Interface Task	Related Configuration File
Define Accounts	Add or update local accounts	Remote site definition file $CMUHOME/rsd
	Add or update remote accounts	Remote site definition file $CMUHOME/rsd
	Configure async connection from Connect:Enterprise	Remote site definition file $CMUHOME/rsd
	Configure data for async transfer	Remote site definition file $CMUHOME/rsd
	Configure bisync connection from Connect:Enterprise	Remote site definition file $CMUHOME/rsd
	Configure data for bisync transfer	Remote site definition file $CMUHOME/rsd
	Configure FTP connection from Connect:Enterprise	Remote site definition file $CMUHOME/rsd

Navigation Bar Heading	User Interface Task	Related Configuration File
	Configure data for FTP transfer	Remote site definition file $CMUHOME/rsd
	Specify FTP PUT Options	Remote site definition file $CMUHOME/rsd
	Specify FTP firewall	Remote site definition file $CMUHOME/rsd
	Specify FTP commands for automatic transfer	Remote site definition file $CMUHOME/rsd
Define Contract	Define contract	configuration file $CMUHOME/
	Apply contract and proxy configuration	configuration file $CMUHOME/
Define Communications	Configure async hardware devices	Communications definition file $CMUHOME/cpd
	Configure async protocol settings and port options	Communications definition file $CMUHOME/cpd
	Configure bisync ARTIC hardware devices	Communications definition file $CMUHOME/cpd
	Configure bisync ARTIC protocol settings and port options	Communications definition file $CMUHOME/cpd
	Configure bisync Cleo hardware devices	Communications definition file $CMUHOME/cpd
	Configure bisync Cleo protocol settings and port options	Communications definition file $CMUHOME/cpd
	Configure FTP communications port options	Communications definition file $CMUHOME/cpd
Define Communications	Define port settings	configuration file $CMUHOME/
Define HTTP Proxy	Define HTTP proxies that handle outbound requests	configuration file $CMUHOME/
	Apply contract and proxy configuration	configuration file $CMUHOME/
Define Configuration	Define system configuration	Mailbox control definition file $CMUHOME/mcd

Navigation Bar Heading	User Interface Task	Related Configuration File
Define Mailbox Access	Define permissions for mailboxes	mbxacl.conf $CMUHOME/med
Define Roles	Define access based on role	New feature; configuration information is in c-tree database
Define Schedules	Define or update automatic transfer schedule	Auto connect definition file $CMUHOME/acd
	Update a schedule for automatic transfer	Auto connect definition file $CMUHOME/acd
	Add contract to an automatic transfer schedule	Auto connect definition file $CMUHOME/acd
	Define data settings for remote account	Auto connect definition file $CMUHOME/acd
	Add a remote account to an automatic transfer schedule	Auto connect definition file $CMUHOME/acd
	Configure async communications connections for schedule	Auto connect definition file $CMUHOME/acd
	Configure async data for schedule	Auto connect definition file $CMUHOME/acd
	Configure bisync communications connections for schedule	Auto connect definition file $CMUHOME/acd
	Configure bisync data for schedule	Auto connect definition file $CMUHOME/acd
	Configure FTP communications connections for schedule	Autoconnect definition file $CMUHOME/acd
	Configure FTP data for schedule	Auto connect definition file $CMUHOME/acd
	Configure FTP PUT options for schedule	Auto connect definition file $CMUHOME/acd
	Configure FTP commands for schedule	Auto connect definition file $CMUHOME/acd
	Select a schedule to run	Auto connect definition file $CMUHOME/acd
Define Security	Define security configuration parameters (Connect:Enterprise Secure FTP)	Security protocol definition file $CMUHOME/spd

Navigation Bar Heading	User Interface Task	Related Configuration File
Define System Policy	Define system password policy	New feature $CMUHOME/etc/ passadmpf.cfg file
Define Account Policy	Define account password policy	New feature $CMUHOME/rsdpolicy
Reset Password	Reset user password by administrator	New feature $CMUHOME/rsd
Change Password	Change user password	New feature $CMUHOME/rsd

Mapping the Configuration Files to the Site Administration Interface

The following table identifies the configuration files and the associated Site Administration user interface tasks:

Configuration File	Related User Interface Task
configuration file $CMUHOME/	Configure communications port options
	Define contract
	Define HTTP proxy
	Apply contract and proxy configuration
Mailbox control file $CMUHOME/mcd	Define system configuration
mbxacl.conf $CMUHOME/med	Define permissions for mailboxes
Security protocol definition file $CMUHOME/spd	Define security configuration parameters (Connect:Enterprise Secure FTP)
Remote site definition file $CMUHOME/rsd	Add or update local accounts
	Add or update remote accounts

Configuration File	Related User Interface Task
	Configure async connection from Connect:Enterprise
	Configure data for async transfer
	Configure bisync connection from Connect:Enterprise
	Configure data for bisync transfer
	Configure FTP connection from Connect:Enterprise
	Configure data for FTP transfer
	Specify FTP firewall
	Specify FTP commands for automatic transfer
Communications definition file $CMUHOME/cpd	Configure async hardware devices
	Configure async protocol settings and port options
	Configure bisync ARTIC hardware devices
	Configure bisync ARTIC protocol settings and port options
	Configure bisync Cleo hardware devices
	Configure bisync Cleo protocol settings and port options
	Configure FTP communications port options
Auto connect definition file $CMUHOME/acd	Define or update automatic transfer schedule
	Update a schedule for automatic transfer
	Add contract to an automatic transfer schedule
	Define data settings for remote account
	Configure async data for schedule
	Configure bisync communications connections for schedule
	Configure bisync data for schedule
	Configure FTP communications connections for schedule
	Configure FTP data for schedule
	Configure FTP PUT options for schedule
	Configure FTP commands for schedule
passadmpf.cfg file $CMUHOME/etc	Define system password

Mapping Site Administration Terms to Configuration Files Terms

The following table identifies the relationship between terms in the Site Administration user interface and parameters in the Connect:Enterprise configuration files:

User Interface Term	Configuration Files Term	Configuration File
Account	Remote site definition file	RSD
Account name	Name of the RSD file	RSD
Add inbound ETX-terminated batches to the mailbox as a single file	CONCATETX	RSD, ACD
After file is retrieved (after GET)	POSTRECEIVE	RSD, ACD
After file is sent (after PUT)	POSTSEND	RSD, ACD
Apply space compression	COMPRESSION	RSD, ACD
identifier for local host		configuration file
identifier for remote trading partner		configuration file
listening port		configuration file
port		configuration file
ASCII end of file character	ASCII_EOF_CHAR	RSD, CPD
ASCII end of file timeout	ASCII_EOF_TIMEOUT	RSD, CPD
At end of session (before QUIT)	SESSIONEND	RSD, ACD
At start of session (after USER and PASS)	SESSIONSTART	RSD, ACD
Authentication	AUTH	SPD
Auto connect	Schedule	ACD
Batch ID	BATCHID	ACD
Baud rate	801C=port:baud (async, bisync)	CPD
Baud rate	ASYNC	CPD
Before file is retrieved (before GET)	PRERECEIVE	RSD, ACD
Before sending file to remote account	PRESEND	RSD, ACD
Board type	HARDWARE	CPD
Business process name	BPNAME	ACD

User Interface Term	Configuration Files Term	Configuration File
Card, ARTIC port number	PORT	CPD
Character set translation table	TRANSLATE	RSD
Check host IP	CHECKHOSTIP	RSD, ACD
Cipher list	CIPHERS	CPD, RSD, ACD
Cipher strength	CIPHER_STRENGTH	SPD
Cipher strength, Define port		configuration file
Cipher strength, Define HTTP/HTTPS Connection		configuration file
Cipher suites	CIPHER_SUITES	SPD
Cipher suites, Define port		configuration file
Cipher suites, Define HTTP/HTTPS Connection		configuration file
Cleo install	CLEOINSTALL	CPD
Cleo parameters	CLEOCMD	CPD
Command timeout	COMMAND TIMEOUT	CPD
Common name validation		configuration file
Communications protocol	PROTOCOL	RSD
Compress Message		configuration file
Concatenate inbound files	CONCATFILES	RSD, ACD
Confirm password	PASSWORD	RSD
Confirm remote password	RMT_PASSWORD	RSD
Connection attempts	PORT_RANGE_RETRIES=	CPD
Connection retry attempts	RETRIES	ACD
Consecutive logon failures	CONSECUTIVE_FAILED_LOGON_FLAG , CONSECUTIVE_FAILED_LOGON_ ATTEMPTS	ASC
Contact remote site	CONTACT	ACD
Data bits	AUXPARMS=databits (async) 801C=databits (bisync)	CPD
Dead-letter mailbox		configuration file
Destination file name	REMOTEFILENAME	RSD, ACD
Device	DEVICE	CPD

User Interface Term	Configuration Files Term	Configuration File
Direct trust certificate file		configuration file
Directory format	DIRFORM	RSD
Enable FTP operations only for Connect:Enterprise	MAILBOXMODEONLY	CPD
Enable interactive mode	INTERACT	CPD
Enable passive mode	PASSIVE	CPD, ACD
Enable port restrictions	PORTRESTRICTION	CPD
Enable SIPS encryption	SIPSENCRYPTION	MCD
End-of-line characters used in automatic transfers	RECORDSEPARATOR	RSD, ACD
Except for these times	EXCEPT	ACD
Exchange algorithm, Configure Inbound Message		configuration file
Exchange algorithm, Configure Outbound Message		configuration file
Exchange certificate, Configure Inbound Message		configuration file
Exchange key certificate		configuration file
Extended deadline for expired certificates		configuration file
Extract never, transmit once	FTP_PUT_OPTIONS=”TO”	ACD, RSD
Extract once, transmit never	FTP_PUT_OPTIONS=”EO”	ACD, RSD
Extract repeatedly, transmit never	FTP_PUT_OPTIONS=””	ACD, RSD
Extract repeatedly, transmit once	FTP_PUT_OPTIONS=”XMIT”	ACD, RSD
Extract repeatedly, transmit repeatedly	FTP_PUT_OPTIONS=”MULTXMIT”	ACD, RSD
Forced expiration date	FORCE_EXPIRATION_DATE_FLAG, FORCED_EXPIRATION_DATE	ASC
FTP client path	CLIENTPATHNAME	CPD
FTP server path	SERVERPATHNAME	CPD
GET command (send in place of MGET)	GETCOMMAND	RSD, ACD
Inbound data format	DATAFORMAT	RSD, ACD

User Interface Term	Configuration Files Term	Configuration File
IP address	FTPSCRIPT	RSD
Keep existing file	SUNIQUE	ACD
Key certificate file		configuration file
Key certificate file	RC_KEYCERT_FILE	SPD
Length of remote file name	RMT_FNAME_LEN	RSD
Line	LINE	CPD
Log on prompt	LOGIN	CPD
Log on timeout	LOGON TIMEOUT	CPD
Log receipt of batch	LOGBATCH	RSD
MAC list	MACS	CPD, RSD, ACD
Mailbox list	SENDID	RSD
Mailboxes trading partner can send to		configuration file
Maximum log file size	MAXLOGFILESIZE	MCD
Maximum number of concurrent sessions	SESSIONS	ACD
Maximum number of log files	MAXLOGFILES	MCD
Maximum password length	MAXIMUM_PASSWORD_LENGTH	ASC
MDN type		configuration file
Message authentication codes, MAC list	MACS	RSD, ACD, CPD
MIME type/subtype		configuration file
Minimum password length	MINIMUM_PASSWORD_LENGTH	ASC
Modem initialization string	MDMCTL (async) MODEMCTRL (bisync)	CPD
Modem phone numbers	PHONE	RSD, ACD
Number of connection attempts	PORT_RETRIES	RSD, ACD
Number of retries for failed attempts		configuration file
Number of saved password used to validate changes	PASSWORD_HISTORY_FLAG PASSWORD_HISTORY	ASC
Outbound batch separation	BCHSEP	RSD, ACD
Outbound data format	AUTOCONVERT	ACD, RSD

User Interface Term	Configuration Files Term	Configuration File
Parity	AUXPARMS=parity (async) 801C (bisync)	CPD
Password	FTPSCRIPT=”PASSWORD”	RSD
Password authentication	PASSWORDAUTH	CPD, RSD, ACD
Password duration in days	DURATION_FLAG, DURATION	ASC
Password prompt	PASSWD	CPD
Password, Define Port		configuration file
Password, Define HTTP/HTTPS Connection		configuration file
Poll daemons	POLLDAEMONS	MCD
Port	801C=port:baud (bisync)	CPD
Port	FTPSCRIPT	RSD
Port range	PORT_RANGE	CPD, RSD, ACD
Port used to access Connect:Enterprise	PORTLISTENER	CPD
Preferred Authentication	PREFERREDAUTH	RSD, ACD
Priority level in queue	ACPRIORITYLEVEL	ACD
Process	daemon	
Protocol	PROTOCOL	CPD
Proxy server URL		configuration file
Public IP address or host name		configuration file
Public key authentication	PUBKEYAUTH	CPD, RSD, ACD
Receive files from	ACRECVDIR	RSD, ACD
Receive inbound files from	ACRECVDIR	ACD, RSD
Records per block Records per block of data	BLOCK	RSD, ACD
Remote host	ADDRESS	RSD, ACD
Remote password	RMT_PASSWORD	RSD
Remote port	FTPPORT	RSD, ACD
Remote trading partner URL		configuration file
Remote user ID	RMT_USER	RSD

User Interface Term	Configuration Files Term	Configuration File
Rename outbound file for backward compatibility	RENAME_FILE	RSD, ACD
Require messages to be encrypted		configuration file
Require messages to be signed		configuration file
Reroute files automatically to other remote accounts.	FTP_PUT_OPTIONS=“TRIGGER”	RSD, ACD
Resource list	RESOURCE	ACD
Retain original message in recipient mailbox		configuration file
Root certificate file	ROOT_CERT_FILE	SPD
Run at these times	WHEN	ACD
Run at these times/Except for these times	WHEN/EXCEPT	ACD
Scan for embedded $ADD cards	SCAN	RSD, ACD
Schedule	Auto connect	ACD
Seconds between connection attempts	PORT_RETRY_WAIT_TIME	RSD, ACD, CPD
Seconds between sessions	STARTDELAY	ACD
Security policy	SECURITY_POLICY	SPD
Security protocol definition file	SECURITY_PROTOCOL_FILE	RSD, ACD, CPD
Security protocol name	Name of the SPD file	SPD
Send buffer size	SENDBUFF	RSD
Send files to, Send outbound files to	ACSENDDIR	RSD, ACD
Send list	SENDID	ACD
Separate concatenated file groups by mailbox ID	MBXSEP	RSD, ACD
Signing algorithm, Configure Inbound Message		configuration file
Signing algorithm, Configure MDN		configuration file
Signing certificate, Configure Inbound Message		configuration file
Signing key certificate, Configure Inbound Message		configuration file

User Interface Term	Configuration Files Term	Configuration File
Specify seconds of inactivity	DISCINTV	ACD, RSD
Specify URL or regular expression URL pattern		configuration file
SSHFTP client name	CLIENTPATHNAME	CPD
SSHFTP server name	SERVERPATHNAME	CPD
SSL client CCC policy	SSL_CLIENT_CCC_POLICY	SPD
SSL server CCC policy	SSL_SERVER_CCC_POLICY	SPD
Stop bits	AUXPARMS=stopbits (async) 801C=bisync (bisync)	CPD
Suppress final interexchange record separator	SFI	RSD, ACD
SYNCcable device	DEVICE	CPD
Time Out		configuration file
Times to re-queue remote resource	REQUEUES	ACD
Transfer sequence	MODE	RSD, ACD
Translated listening port		configuration file
Truncate trailing blanks before transmission	TRUNC	RSD, ACD
Trust host key	TRUSTHOSTKEY	RSD, ACD
Trusted certificates file		configuration file
Turn on password length control	PASSWORD_LENGTH_FLAG	ASC
Type of protocol at remote site	TYPE	RSD
Use DNS	USEDNS	CPD
Use hardware flow control	FLOWCONTROL	CPD
User ID		configuration file
User ID, Define Port		configuration file
User ID, Define HTTP/HTTPS Connection		RSD
User mailbox access	MAILBOX_LIST	RSD
Valid mailbox list	VALIDIDLIST	MCD
Wait for Data Set Ready (DSR) signal before sending initialization string	WAITDSR	CPD

User Interface Term	Configuration Files Term	Configuration File
Wait to re-queue failed remote connection	INTERVAL	ACD
Working directory for Cleo 3780Plus	CLEOWORK	CPD

Mapping the Site Administration Interface Tasks to Related Utilities

The following table identifies the relationship between tasks in the Site Administration user interface and the Connect:Enterprise utilities:

User Interface Task	Connect:Enterprise Utility
List batches in repository	cmulist
Delete batches from repository	cmudelete, cmuerase
Update batches in repository	cmustatus
View batch details	cmulist
Generate a server report	cmureport
Extract batch from repository	cmuextract
Generate a remote account report	cmureport
Generate a schedule report	cmureport
Generate report	report
Change user password	ceupasswd
Display RSD policy files	ceupassrpt
Create and maintain password policy files and RSD policy files	cuepassadm

Index

Symbols

Numerics

A

B

C

L

M

N

P

R

S

T

U

V

W