Connect:Enterprise®
for z/OS

Administration Guide

Version 1.4

Sterling Commerce
An IBM Company

Connect:Enterprise for z/OS Administration Guide

Version 1.4

First Edition

Sterling Commerce, Inc.
4600 Lakehurst Court Dublin, OH 43016-2000 *
614/793-7000

Contents

Chapter 4 Configuring ODF Records for SNA Connections 65

Chapter 5 Configuring ODF Records for FTP Connections 99

Chapter 16 Diagnostics 371

Appendix A Automating Connect:Enterprise 383

Appendix B Worksheet for Remote Sites 393

Glossary 397

Index 409

Chapter 1

About Connect:Enterprise

Connect:Enterprise for z/OS is an online communications and data repository system that enables data transmission between a host computer and remote terminals or computers. Connect:Enterprise provides a way to collect, distribute, and track data, while protecting the host from unauthorized access.

Transmitting and Collecting Data

Connect:Enterprise for z/OS collects and transmits data between the host computer and remote terminals, applications, or computers through the *data repository,* which is designed on the store and forward model. Like a voice mail system, the data repository consists of individual *mailboxes,* or directories*,* where data files are stored for future processing by the host or remote site. The Connect:Enterprise administrator assigns mailboxes and controls access to the mailboxes through Connect:Enterprise for z/OS user IDs and passwords. After a communications session is established between Connect:Enterprise for z/OS and a remote site, either the host or the remote users can store, retrieve, or monitor data files in the mailboxes to which they have access. Both the host computer and the remote sites can initiate data collection and distribution. A communications session with the Connect:Enterprise for z/OS repository can be initiated using the FTP, SNA, or BSC protocol.

Connect:Enterprise for z/OS collects data files from remote sites for a central host site. For example, Connect:Enterprise can gather data generated by a database application for one remote site, then extract the data at the host site for use by a local application.

Connect:Enterprise for z/OS distributes data files from the host to one or more remote sites. The host site can automatically transmit output reports or data to remote sites. For example, if a company needs to send the latest sales report to its 25 regional sales offices, it can either send the report at a predetermined time to its sales offices or deposit the report in the repository and flag the report for transmission to the offices. The remote offices connect to the repository, obtain a listing of the repository contents, and request transmission of reports to their sites.

Connect:Enterprise for z/OS also enables you to schedule automated data collection and transmission between the host and an unattended remote site using the Auto Connect feature. You can schedule automated sessions by time of day, day of the week, day of the year, or you can initiate an Auto Connect session by issuing a host site operator command when data is ready for transmission. An Auto Connect activity report is available after the Auto Connect session finishes.

Connect:Enterprise for z/OS Terminology

The following table defines the concepts and terminology used with this product.

Term	Connect:Enterprise for z/OS Definition
Remote site	Remote terminal, application, or computer which is configured to initiate a communications session with the data repository.
Host site	Connect:Enterprise for z/OS server on which the data repository resides.
Data repository	Collection of individual mailboxes, or directories, where data files are stored for future processing by the host or remote site.
Batch	Data file residing in a mailbox of the repository on the Connect:Enterprise host computer. When a batch is added to the repository, it is assigned a unique number (from 1 to 9,999,999).
Mailbox ID	An identifier that defines the repository associated with a batch. Remote users access the contents of the VSAM batch files, where the batches are stored, using the Mailbox ID. A user can have access to a single mailbox (individual mailbox), a group mailbox (accessible by multiple users), and multiple mailboxes (accessible by a single user or trading partner).
Batch ID (BID) or user batch ID	A description of the batch. It is also referred to as the user batch ID because it is assigned by the user.
Auto Connect	Unattended, scheduled communications session initiated by the Connect:Enterprise for z/OS repository to distribute or collect data.
Remote-initiated session (Remote connect)	Unsolicited communications session with the Connect:Enterprise for z/OS repository initiated by a remote site.

Remote Site Capabilities

When you initiate a communications session with the Connect:Enterprise data repository from a remote site, you can perform the following functions:

Function	Description
Add batches	Add batches to the data repository by establishing a session with Connect:Enterprise. Each transmission generates one or more batches. All batches transmitted from a remote site are marked with the mailbox ID supplied or with the remote name of that site. The originating site can also flag batches as transmittable, making them available for transmission requests by other remote sites.
Request batches	Request batches from the data repository by establishing a session with Connect:Enterprise and requesting a batch by the appropriate batch identifier. Batches requested this way are not deleted. They remain in the data repository and are still available for transmission to other remote sites.
Delete batches	Instructs the Connect:Enterprise repository to flag a batch as deleted. The batch is logically deleted and not transmitted to remote sites. The batch is displayed in directory listings but is flagged as deleted. Batches are physically removed only by using the ERASE utility, which is not available to remote sites.
List directory of batches	Request a list of all batches in the data repository marked with either their mailbox ID or a given batch ID.

Connect:Enterprise for z/OS Communications

Connect:Enterprise runs on a host mainframe using the Virtual Telecommunications Access Method (VTAM), Basic Telecommunications Access Method (BTAM), or Transmission Control Protocol/Internet Protocol (TCP/IP), and enables data transmission between the host and remote terminals or computers. Connect:Enterprise supports data transmissions between the host and remote sites using the following types of protocols:

- SNA data transmissions between mainframes using VTAM and various remote devices
- BSC data transmissions between mainframes using BTAM and various other computers
- FTP data transmissions between remote FTP clients and the Connect:Enterprise FTP server
- FTP data transmissions between remote FTP servers and the Connect:Enterprise FTP client
- FTP SSL data transmissions between remote FTP clients and the Connect:Enterprise FTP server
- FTP SSL data transmissions between remote FTP servers and the Connect:Enterprise FTP client

Also, Connect:Enterprise Gateway provides data transmission between the Connect:Enterprise host computer and ASYNC, BSC, and FTP remote sites.

The types of communication lines that you use determine how you connect to the data repository. Binary synchronous lines are typically either switched (dialup) or nonswitched (dedicated or leased) lines. A switched line is a temporary connection between two sites for the duration of the session only. A nonswitched line is a permanent connection between sites that does not require dialup to start the communications.

Security

You can implement security in the Connect:Enterprise system at various points in the processing. The security implementation at the Connect:Enterprise host affects the requirements at the remote sites.

Ensure that you have complete instructions from your host site administrator regarding the security measures that are implemented on your system, the functions that are available to you, and the information you need to access the data repository. Refer to the glossary for security terms and definitions. The following table lists components and topics affecting security and where you can find additional information.

Component or Topic Related to Security	**Connect:Enterprise for z/OS Documentation**
Security exits	*Connect:Enterprise for z/OS Application Agents and User Exits Guide* *Connect:Enterprise for z/OS Remote User's Guide*
ODF parameters and configuring for security	*Connect:Enterprise for z/OS Administration Guide* *Connect:Enterprise InterConnect Option for z/OS User's Guide*
Logon and batch function security	*Connect:Enterprise for z/OS Administration Guide*
Authenticating user IDs and passwords	*Connect:Enterprise for z/OS Administration Guide* *Connect:Enterprise for z/OS Application Agents and User Exits Guide* *Connect:Enterprise InterConnect Option for z/OS User's Guide* *Connect:Enterprise for z/OS Remote User's Guide*
Encryption	*Connect:Enterprise for z/OS Administration Guide* *Connect:Enterprise for z/OS Application Agents and User Exits Guide* *Connect:Enterprise for z/OS Remote User's Guide*
Security for FTP, SNA, and BSC connections	*Connect:Enterprise for z/OS Administraton Guide* *Connect:Enterprise for z/OS Remote User's Guide*
Connect:Enterprise Security Interface	*Connect:Enterprise for z/OS Administration Guide* *Connect:Enterprise for z/OS Remote User's Guide*
Connect:Direct security	*Connect:Enterprise InterConnect Option for z/OS User's Guide*

Connect:Enterprise for z/OS Components

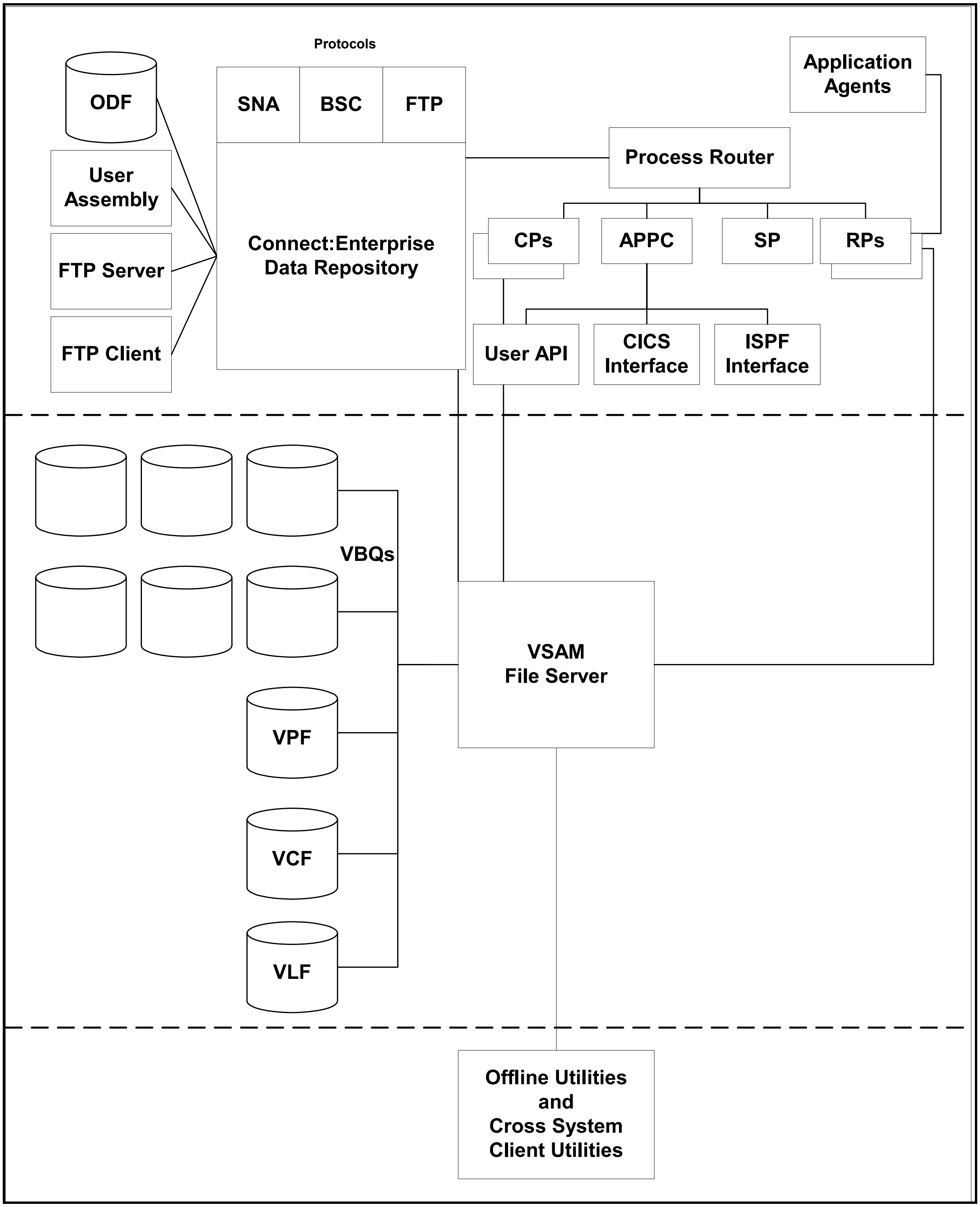
Connect:Enterprise has three major components:

✦ Data repository or Connect:Enterprise for z/OS online system

✦ Virtual Storage Access Method (VSAM) file server

✦ Offline utilities

The following diagram illustrates these components.



Data Repository

The data repository transmits and collects data from BSC, FTP, and SNA sites. The repository handles all session activity and accepts service requests from the console, the user API, the ISPF interface, the CICS interface, and the Connect:Enterprise FTP server.

The Connect:Enterprise data repository consists of the following components:

Component	Description
Options Definition File (ODF)	The ODF contains control information such as passwords, security information, connection definitions, logical unit pooling, remote site definitions for SNA and FTP sites, and signon records and BTAM ID verification for BSC sites.
User Assembly	The User Assembly contains macros that define BSC lines to Connect:Enterprise. This component is used only by BSC connections.
FTP Server	The FTP Server enables remote FTP client sites to access, retrieve, and send data to Connect:Enterprise through a subset of the standard FTP commands.
FTP Client	The FTP Client enables FTP communications between FTP servers implemented on any platform.
Command Processors (CPs)	The CPs process command requests from the ISPF interface, the CICS interface, and the user API.
Advanced Peer-to-Peer Communications (APPC)	The APPC server provides LU6.2 communications with the ISPF interface, the CICS interface, Cross System Client, and the user API.
Process Router (PR)	The PR routes transactions among the repository, the CPs, the APPC server, and RPs.
Application agents	Application agents are commands that you can use to customize and automate Connect:Enterprise processing.
Rules Processors (RPs)	The RPs process requests from the application agent interfaces.
Scheduler Processor (SP)	The SP manages timer intervals, which schedule when events are triggered.

The VSAM File Server and VSAM Batch Files

The VSAM file server processes requests from the Connect:Enterprise data repository to read and write data to the VSAM batch files. The VSAM batch files consist of the VSAM Pointer File (VPF), VSAM Control File (VCF), VSAM batch queues (VBQ), and VSAM log files (VLF). The following table describes the functions of these files.

VSAM Batch File	Function
VSAM Pointer file (VPF) VSAM Control file (VCF)	Indexes and controls the information regarding the batches, such as mailbox ID, status, and location of the batch data within the VSAM batch queues.

VSAM Batch File	Function
VSAM Batch Queue (VBQ)	Multiple files that contain the actual batch data.
VSAM Log files (VLF)	Contain the record of communications sessions and offline utility activity.

The VSAM file server resident task must be active for the repository and offline utilities to run.

Offline Utilities

Offline utilities enable you to maintain VSAM batch files by performing such tasks as adding and extracting batches from the VBQs and reporting. Offline utilities access the VSAM file server to perform these tasks.

Offline utilities must reside in the same logical partitioning (LPAR) as the VSAM file server. However, you can run a subset of the offline utilities, called the Cross System Client Utilities, from a separate LPAR.

Connect:Enterprise for z/OS Documentation

See *Connect:Enterprise for z/OS Release Notes* for a complete list of the product documentation.

About This Guide

Connect:Enterprise for z/OS Administration Guide is for programmers and network operations staff who maintain the product.

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

Notational Conventions

The Connect:Enterprise for z/OS documentation uses certain notational conventions. This section describes the conventions used in this guide.

Convention	Description
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.
UPPERCASE and lowercase Letters	A statement, command, or parameter in uppercase letters followed by lowercase letters indicates an alternative to typing the entire command. For example, `SELect PROCess` means that you need only type `SEL PROC` for the command to be valid.

Convention	Description
lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, `PNODE=primary-node-name` indicates that you must provide the name of the primary node.
Bold Letters	Bold print in syntax boxes indicates commands and required parameters. For example, **`NAME=filename`** indicates that the parameter *NAME* is required.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, `FTP=Yes|No` specifies that the default for *FTP* is *NO*.
Vertical Bars (|)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example `HOLD=Yes|No|Call` specifies that *Yes* or *No* or *Call* is valid.
Brackets [	Brackets indicate that information is optional. For example, `STARTT=([date|day][,hh:mm:ssXM])` indicates that you can specify either a date or a day, a date or a day plus a time, or just a time.
Italics	Italic letters are placeholders for information you must provide. Italic font also indicates book, chapter, and section titles and is used for emphasis in the text.
`Monospaced characters` (characters of equal width)	`Monospaced characters` represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.

Chapter 2

Understanding the Options Definition File

The Options Definition File (ODF) contains product options specific to your site. You must define the ODF to initialize Connect:Enterprise. The installation guide provides sample ODFs to use to verify the installation because creating your site-specific ODF can be time-consuming. After you verify the installation, you should use the information and procedures in this guide to create and verify your site-specific ODF. You update the ODF when you add or delete a remote site, change the settings for a remote site, or change the Connect:Enterprise default setup.

This chapter explains the function and structure of the ODF, illustrates the example ODF that accompanies the product, summarizes the tasks required to configure the ODF, defines general rules for creating the ODF, describes how to create and verify the ODF, and summarizes the ODF records and parameters according to the component or functionality they are used to configure.

ODF Function and Structure

The ODF is either a sequential file or a member in a partitioned data set containing 80-byte records. ODF records are read and verified during Connect:Enterprise initialization. Each 80-byte record is scanned for valid keywords and data. Keywords and data cannot be continued across multiple 80-byte records.

Invalid data in the ODF records terminates Connect:Enterprise and sends an error message to the console. An example of invalid data is edit sequence numbers in columns 73–80 for ODFs created with a TSO/ISPF text editor.

ODF Records and Function

The ODF consists of several sections delimited by special control records. Depending on your setup and how you implement Connect:Enterprise, you may not require all the sections of the ODF. To see the sample ODF provided with Connect:Enterprise, see the ODFDEF member of the ENTPRS.EXAMPLE library. The following table describes the function and position of the ODF records, and the conditions under which they are required.

ODF Record	Function	Position
*OPTIONS	Defines the default settings for parameters that are used to initialize and define the default behavior of the product, configure the default values for its resources, and identify and define default values for parameters for SNA, FTP, and BSC connections.	Required always; must be the first record in the ODF.
*SECURITY	Contains a list of the Mailbox IDs valid for your system. Use to control access to batches by remote SNA and BSC sites. When you implement batch security, Connect:Enterprise for z/OS verifies that remote SNA and BSC connections use a Mailbox ID listed in the *SECURITY record before action is taken. FTP connections are not subject to *SECURITY validation.	Required if SECURITY=BATCH is set in the *OPTIONS record, otherwise optional.
*CONNECT	Implements the Auto Connect function (host-initiated connection), defines the name of the Auto Connect list and type of protocol used for the connection, lists the remote site, or sites, that the host contacts, the time of day to activate the transmission, and additional site-specific processing options. To schedule specific days or exceptions for processing, you must also include the *CALENDAR record.	Optional. No required position.
*CALENDAR	Defines schedules and exceptions by day of week, date, or both for activating and deactivating time-initiated Auto Connect sessions.	Optional. No required position.
*REMOTES	Identifies the name and type of SNA and FTP sites that can establish a session with the Connect:Enterprise repository and defines default site-specific settings for connections.	Required for SNA and FTP sites. No required position.
*POOLS	Identifies a pool of logical unit names used when an Auto Connect session to SNA remote sites is initiated. Replaces LUNAME or RMTACB parameters in the *REMOTES record.	Optional for SNA sites. No required position.
*IDVER	Defines remote IDs and host IDs to exchange when BTAM ID verification is specified for BSC lines.	Optional for BSC sites. No required position.
*SIGNON	Identifies the valid SIGNON formats a remote site may send when establishing a connection with the host.	Optional for BSC sites. Must be the last record in the ODF, if present.

Caution: Because the records in the *SIGNON section can be free-form, ensure that the *SIGNON section is the last section of the ODF if you use this record. If the *SIGNON record precedes another record (*REMOTES, *CONNECT, *CALENDAR) and that section identification record is accidentally deleted, the records for that section are appended to the *SIGNON section and not validated by the ODF verification step, but no error is reported at startup. However, those records will be considered missing and unavailable for use.

Preparing to Define ODF Records

Before you begin defining the ODF for your site:

✦ Review *ODF Configuration Tasks* on page 23.

✦ Review the worksheet in Appendix B, *Worksheet for Remote Sites*.

✦ Review *General Rules for Creating the ODF* on page 24.

✦ Review *Summary of ODF Records and Parameters* on page 27.

ODF Configuration Tasks

The chapters devoted to configuring the ODF are organized according to the major tasks required to create this file. The first major task is to define the default values for the parameters that control the Connect:Enterprise for z/OS system resources. The tasks related to defining remote sites and Auto Connects are organized according to protocol; that is, the task flow assumes that after you define the values for the Connect:Enterprise for z/OS resources, you configure the records required for connections from remote sites to the Connect:Enterprise for z/OS repository based on whether the SNA, BSC, or FTP protocol is used for the communications session. Configuring the information required for connections from remote sites to the Connect:Enterprise for z/OS repository will aid you in configuring Auto Connects for the different protocols.

The following table lists the major tasks and records used to configure the ODF and the chapters that describe the format, rules, and parameters for the records.

Note: Your site-specific ODF is incomplete until you define all the parameters and records required for your system, comment out or omit those parameters not used, and execute ODF verification.

Task	ODF Record	Related Chapter
Configure ODF parameters for Connect:Enterprise system resources	*OPTIONS	Chapter 3, *Configuring *OPTIONS Record for System Resources*
SNA Connections		
Set default values for SNA connections	*OPTIONS	Chapter 4, *Configuring ODF Records for SNA Connections*
Define valid Mailbox IDs and specify to check for valid LU names if necessary	*SECURITY	
Define a record for each remote site that can initiate a connection to the Connect:Enterprise for z/OS repository	*REMOTES	
Create pool of logical unit (LU) names used to initiate SNA Auto Connects	*POOLS	

Task	ODF Record	Related Chapter
Configure SNA Auto Connect list	*CONNECT	
Add SNA site or sites to Auto Connect list	SNA Remote Site Specification	
FTP Connections		
Configure FTP default values for client and server connections, including security	*OPTIONS	Chapter 5, *Configuring ODF Records for FTP Connections*
Configure ODF parameters for connections to remote FTP servers	*REMOTES record for FTP server	
Configure ODF parameters for connections from remote FTP client sites, including anonymous and generic FTP remote site definitions	*REMOTES record for FTP client	Chapter 5, *Configuring ODF Records for FTP Connections*
Configure FTP Auto Connect list	*CONNECT	
Add FTP site to Auto Connect list	FTP Remote Site Specification	
BSC Connections		
Set default values for BSC connections	*OPTIONS	Chapter 6, *Configuring ODF Records for BSC Connections*
Define valid Mailbox IDs if necessary	*SECURITY	
Define host ID and remote ID to exchange if BTAM ID verification is specified for any lines	*IDVER	
Define signon data format, if necessary	*SIGNON	
Configure BSC Auto Connect list	*CONNECT	
Add BSC site to Auto Connect list	BSC Remote Site Specification	
Auto Connect Lists		
Create Auto Connect schedule, or schedules, and specify in the appropriate Auto Connect list	*CALENDAR	*Chapter 7, Configuring *CALENDAR Records*

General Rules for Creating the ODF

When you create the ODF, you must observe some general rules as well as rules specific to each type of record. Record-specific rules are listed in the section that describes the parameters for each record type. Observe the following general rules:

- Indicate comments with an asterisk in columns 1 and 2 (**); comments can occur anywhere in the ODF.
- The *OPTIONS record is required and must be the first record in the ODF.

- The *REMOTES record is required for FTP and SNA sites.
- The *SIGNON record must be the last record of the ODF.
- At initialization, Connect:Enterprise scans the ODF input and terminates the system initialization if it detects input errors. Input errors are generated for the presence of optional parameters or sections that are not used for your implementation. You can avoid errors at initialization that result from the presence of unused, optional parameters and sections in the following ways:
  - You can delete any records containing unused parameters when you create your site-specific ODF.

    Note: To ensure that you do not inadvertently delete records and parameters that you need, review *Summary of ODF Records and Parameters* on page 27 before you delete any of the records and parameters from the sample ODF.

  - You can comment out any unused optional parameters and records by placing an asterisk in columns 1 and 2 (**) of the line on which they appear.

    Note: Commenting out unused parameters and records may be more efficient because it ensures that you do not delete sections or parameters that you may need later.

- The *OPTIONS, *SECURITY, *CONNECT, *CALENDAR, *REMOTES, *POOLS, *IDVER, and *SIGNON record type name must begin in column 1; all other text on the same line is ignored.
- Required keywords must precede optional keywords.

  Sample ODF records show required keywords in the correct position. ODF parameter tables in chapters 3 through 7 list required parameters first in bold font; positional parameters are listed in the correct order; and optional parameters are listed in alphabetical order.
- Although a parameter may be specified more than one time in the ODF, only the last value is used.

Controlling Attributes of Communications Sessions Using ODF Records

Connect:Enterprise for z/OS communications sessions can be host-initiated connections to remote sites (Auto Connect sessions) and remote-initiated sessions to the Connect:Enterprise for z/OS host. Remote-initiated connections are unsolicited connections with Connect:Enterprise. No action by the Connect:Enterprise host causes these connections to occur. For these connections to be successful, the Connect:Enterprise ODF must be configured with parameters to accept the remote-initiated connections, and the remote site must supply connection parameters that are acceptable to the Connect:Enterprise host.

Parameters set in the *OPTIONS record enable and define the most generic and generally applicable values for communications sessions using the SNA, FTP, and BSC protocols. The *OPTIONS values define the attributes of host-initiated and remote-initiated communications sessions if no overriding equivalent parameters are defined in other ODF records.

Creating and Verifying Your Site-Specific ODF

The following procedure describes the steps required to create and verify your ODF. This procedure assumes that you modify the OPTDEF example member ODF file that is distributed with the product to configure the ODF records necessary for your implementation. Use this procedure as a checklist to ensure that you have completed all the tasks required to create and verify the ODF.

To create your site-specific ODF:

1. Edit the OPTDEF member of the ENTPRS.EXAMPLE library.
2. Supply a valid job card.
3. Change `//SYSUT2 DD DSN=ENTPRS.OPTFILE,DISP=(NEW,KEEP),UNIT=XXXX,VOL=SER=NNNNNN` as follows:
   a. Replace ENTPRS.OPTFILE with the DSNAME of your choice.
   b. Supply the UNIT and VOLSER number for your system.
4. Define the *OPTIONS record parameters for your Connect:Enterprise for z/OS system resources using the parameter definitions in Chapter 3, *Configuring *OPTIONS Record for System Resources*.
5. Depending on the requirements of your system, configure the records and parameters described in the following chapters:
   - Chapter 4, *Configuring ODF Records for SNA Connections*
   - Chapter 5, *Configuring ODF Records for FTP Connections*
   - Chapter 6, *Configuring ODF Records for BSC Connections*
   - Chapter 7, *Configuring *CALENDAR Records*
6. Comment out or delete any unused records and parameters.

Note: The *REMOTES section is required for remote FTP and SNA sites.

7. If you have not already done so, start the VSAM file server from the system console by issuing the following command, where *procname* is the name of the VSAM file server startup process you created during the installation:

```
S procname
```

8. Verify the ODF using the instructions in Chapter 8, *Creating the Connect:Enterprise Startup Task*. You can verify the ODF by starting Connect:Enterprise for z/OS with or without the Verify option with the following results:
   - If you start Connect:Enterprise for z/OS without using the Verify option and your ODF contains errors, the initialization fails with the completion code USER=253 and console messages describe the error conditions; if no ODF errors are present, Connect:Enterprise for z/OS is initialized and continues to execute.

- When you use the Verify option, Connect:Enterprise for z/OS stops execution after the verification.

9. If necessary, review the error messages, correct the ODF errors, and repeat Step 8 on page 26.
10. Secure the ODF from unauthorized use if it contains sensitive information, such as the system password and Mailbox IDs for batch security.

The ODF can also be defined in your Connect:Enterprise job as SYSIN records (//OPTDEF DD *).

Summary of ODF Records and Parameters

The tables in this section summarize the ODF records and parameters according to their use in configuring the system resources and SNA, BSC, and FTP host-initiated and remote-initiated connections.

Connect:Enterprise for z/OS System Resources

The following table summarizes the *OPTIONS record parameters that are required to initialize and define the default behavior of the product; configure global default values for its resources, including the CICS, ISPF, and ICO interfaces, communications, Auto Connect functionality, and security; and customize Connect:Enterprise for z/OS. Bold indicates required parameters.

Component/Resource	Function	*OPTIONS Record Parameter
System	Required for initialization	**APDSN=data.set.name**
	Required for initialization	**VPF='data.set.name'**
	Required for initialization	**DEFAULT_MODE=BID24|BID64**

	Configure VSAM file server	VBQPCT=50 | nn
		VBQROTAT=1 | nn
		VLFPCT=50 | nn
		VLFROTAT=1 | n

	Customize prompt	CMB001I='xxx...xxx'

	Performance	MAXCP=nn | 2
		MAXRP=nn | 2
		BROWSE_AUTOCLEAN_INTERVAL=60 | nnnnn
		BROWSE_DATASPACE_COUNT_MAX=20 | nnn
		BROWSE_DATASPACE_SIZE_MAX=524288 | nnnnnn
		BROWSE_SESSION_COUNT_MAX=40 | nnnn
		BROWSE_SESSION_RETIREMENT_AGE=300 | nnnn

Component/Resource	Function	*OPTIONS Record Parameter

VSAM file server	Define default behavior to release a resource that is assigned to a specific task	DALLOC_RETRY_INTERVAL=30–-3600
		DALLOC_VBQ_INUSE=FAIL |RETRY
		DALLOC_VBQ_STOUTL=ALLOW | DISALLOW
		DALLOC_VLF_INUSE=FAIL |RETRY
		DALLOC_VLF_STOUTL=ALLOW | DISALLOW

APPC, CICS, ISPF, ICO interfaces	Activate interfaces	APPC=YES | NO
		APPCAPPL=xxxxxxxx
		APPCPLSZ=nnnn | 316
		CSC_DEFAULT_REPORTS_FORMAT=1|1X|2
		ICO_DEFAULT_REPORTS_FORMAT=1|1X|2

CICS interface	Identification	CICSAPPL=xxxxxxxx
		CICSMODE=xxxxxxxx
		CICSTR1=xxxx | CM62

z/OS MODIFY command interface	Enables using Connect:Enterprise for z/OS $$ commands in the MODIFY interface	MODIFY=YES | NO | RESP

Console	Logging/messages	CONSLOG=YES | NO
		CONSOLEDESC=nn | 07
		CONSOLEROUT=nn | 01
		COUNT=RECORD | BLOCK

Connect:Enterprise for z/OS Security Interface	Activates the security interface for offline utilities, Connect:Enterprise for z/OS user interfaces, global security, protocol-specificsecurity, and Connect:Enterprise for z/OS component security (ICO, ISPF)	MBXHLQ=xxxxxxxx | MAILBOX
		MBXNAME=xxxxxxxx | MAILBOX
		MBXSECURE=LOGON | BATCH | ALL | WARN | OFF
		CSCSECURE=LOGON | BATCH | ALL | WARN | OFF
		APISECURE=LOGON | BATCH | ALL | WARN | OFF
		ICOSECURE=LOGON | BATCH | ALL | WARN | OFF
		BSCSECURE=LOGON | BATCH | ALL | WARN | OFF
		SNASECURE=LOGON | BATCH | ALL | WARN | OFF
		FTPSECURE=LOGON | BATCH | ALL | WARN | OFF
		STLSECURE=BATCH | WARN | OFF
		UIFSECURE=LOGON | BATCH | ALL | WARN | OFF
		PASSWORD=xxxxxxx
		PASSWORD_CASE=UPPER | MIXED | BOTH

Component/Resource	Function	*OPTIONS Record Parameter
Auto Connect	Specifies default value for queuing an Auto Connect list if no value is specified by the ACQUEUE parameter in the *CONNECT record	ACQDEFAULT=YES | NO

Application agent processing	Customize Connect:Enterprise for z/OS	MAXRP=nn|2
		RULES=YES | NO
		RULES_IR=YES | NO
		RULES_CN=YES | NO
		RULES_CN_PREFIX=xx
		RULESCON=xxxxxxxx
		RULESEOB=xxxxxxxx
		RULESLOG=xxxxxxxx
		RULESSCH=xxxxxxxx
		RULESWKT=xxxxxxxx
		RULES_RECURSION_MAX=nnnnnnnnnn | 5

User exits	Customize Connect:Enterprise for z/OS	XAPPCSEC=xxxxxxxx
		XAPPCWI=xxxxxxxx
		XAPPCWT=xxxxxxxx
		XENDOFB=xxxxxxxx
		XEOBVER=2
		XINIT=xxxxxxxx
		XINPUT=xxxxxxxx
		XLOG=xxxxxxxx
		XOUTPUT=xxxxxxxx
		X_SECURE=STSECFTP
		XSECUR1=xxxxxxxx
		XSECUR2=xxxxxxxx
		XTERM=xxxxxxxx

Reporting/ troubleshooting	Collect information for solving problems and monitoring system	DIALOG_FTP=NO|*| remote1|remote1*,.....| remote8|remote8*
		STOUTL_DEFAULT_REPORTS_FORMAT=1 | 1X | 2
		SUMMARY=ONLY | ANY | FINAL
		SYSOUTCLASS=X | x
		TCPSCH=xxxxxxx
		TRACE=ALLTP|APO|APQ|CP|EXITS|PR|SNA|TCPSCH| VA2C | VSAM | RPEOB | RPLOG | RPWKT | RPCON | RPSCH
		TRACE_FTP *
		TRACEID=xxxxxxxx

ODF Records and Parameters for Configuring SNA Connections

The following table summarizes the ODF records and parameters use to configure SNA connections. Bold indicates required parameters.

ODF Record	SNA Parameter
*OPTIONS	Defines global values for SNA connections.
	APPLID=xxxxxxxx | ENTPRS
	LOGONMSG=’xxx...xxx’ | NO
	MAXRWAIT=HH:MM:SS,C
	SCINCOR=YES | NO
	SECURITY=BATCH | LOGON
	SNA_DEFAULT_$$DIR_FORMAT=BID24|BID64
	VSESSLIM=08 | nn
	VTAM=YES
*SECURITY	Secures batches by verifying Mailbox IDs for remote connections from SNA sites without implementing Connect:Enterprise for z/OS Security Interface. Required if SECURITY=BATCH in *OPTIONS record.
	ID=xxxxxxxx, ID=xxxxxxxx, ID=xxxxxxxx,
*REMOTES	Defines the name, type, and default site-specific settings for remote SNA sites. Values defined in this record take precedence over global settings defined in the *OPTIONS record.
	NAME=xxxxxxxx
	TYPE=LU1RJE
	BCHSEP=OPT3
	BLKSIZE=nnnn | RUSIZE
	COMPRESS=YES | NO
	CONSOLE=YES | NO
	DISCINTV=nnnn
	FMH=YES | NO | X25 | NPP | IE
	GEISMSG xxxx...xxxx
	LOGMODE=xxxxxxxx
	LUNAME=xxxxxxxx | [,xxxxxxxx,...]
	MEDIA=CN | PR | PU | EX | BX
	PASSWORD_CASE=UPPER | MIXED | BOTH
	POOL=xxxxxxxx

ODF Record	SNA Parameter
	QSESS=YES | NO
	RMTACB=xxxxxxxx
	SC=YES | NO | SPC
	TRANSPAR=Y | N
	TRUNC=YES | NO
	USERDATA=‘xx....xx’
*POOLS	Enables you to specify multiple LU names to use for establishing an Auto Connect session with a remote SNA site for which multiple LU names represent the same physical location.
	NAME=xxxxxxxx
	LU=xxxxxxxx
*CONNECT	Defines the name, type, and characteristics of Auto Connect lists.
	LISTNAME=XXXXXXXX
	TYPE=LU1RJE
	ACQUEUE=Y | N
	ACSESS#=01 | nn
	CALENDAR=xxxxxxxx
	DELAY=01 | nnnn
	DISCINTV=15 | nnnn
	MAXRMT#=nn
	NOBATCH=C | NC
	RETRY=0 | nn
	TIME=hh:mm | [,hh:mm, ...]
SNA Remote Site Specification	Specifies the remote SNA site or sites to contact and additional options for the site. Values defined in these records override global SNA settings defined in the *OPTIONS record and the site-specific values set in the *REMOTES record.
	REMOTE_NAME
	BCHSEP=OPT3
	CMP=Y | N
	MEDIA=CN | PR | PU | EX | BX
	ONEBATCH=YES | NO (OB=Y | N)
	TRUNC=N | Y
	BEGINLIST=xxxxxxxx

ODF Record	SNA Parameter
	IDLIST=xxxxxxxx | [,xxxxxxx,...]
	ENDLIST=xxxxxxxx

ODF Records and Parameters for Configuring FTP Server and Client Connections

The following table summarizes the ODF records and parameters used to configure default values for FTP server and client connections, including those related to implementing the SSL or TLS security protocol. Bold indicates required parameters.

Note: Connect:Enterprise for z/OS supports both the TLS (Transport Layer Security) and SSL (Secure Sockets Layer) protocols. Throughout this chapter, the phrase SSL is used to describe both the SSL and TLS protocols.

ODF Record	FTP Parameter
*OPTIONS	Defines global values for FTP connections, including security.
	FTP=NO | YES
	FTP_AC_SCRIPT_DEFAULT=xxxxxxxx | *blank*
	FTP_ALLOW_GETBYNBR_DFLAG_DEFAULT=NO|YES
	FTP_CONNECT_INTERVAL= 0060 | nnnn
	FTP_CLIENT_PASV_DATA_IPADDR=R227|CPADDR
	FTP_DEFAULT_CLIENT_BCHSEP_NONE_FILENAME_FORMAT=BID24|BID64
	FTP_DEFAULT_CLIENT_BCHSEP_OPT3_FILENAME_FORMAT=BID24|BID64
	FTP_DEFAULT_CLIENT_CONTROL_PORT_RANGE=nnnnn-nnnnn, nnnnn-nnnnn
	FTP_DEFAULT_CLIENT_DATA_PORT_RANGE=U | nnnnn-nnnnn, nnnnn-nnnnn
	FTP_DEFAULT_CLIENT_LOCDIRFORM=BROWSER|BROWSER64| MBOX_CLIENT|MBOX_CLIENT64| MBOX_ZOS|MBOX_ZOS64|$MIBNSDFXY| UNIX|UNIX64
	FTP_DEFAULT_CLIENT_REMOTE_FILENAME_LENGTH=SHORT|LONG|LONG64
	FTP_DEFAULT_CLIENT_SCAN=NO | YES | ALL
	FTP_DEFAULT_DIALOG_TRACE_LRECL=136 | nnnnn
	FTP_DEFAULT_DISCTINV=900 | 0 | 3600
	FTP_DEFAULT_KIRN=YES | NO

ODF Record	FTP Parameter
	FTP_DEFAULT_PORT_RETRIES=nn | 0
	FTP_DEFAULT_PORT_RETRY_WAIT_TIME=nnn | 030
	FTP_DEFAULT_RECEIVE_OPTION_RENAME=FIRST 24|LAST24|FIRST64|LAST64
	FTP_DEFAULT_RIFS=YES | NO
	FTP_DEFAULT_SERVER_BCHSEP_NONE_FILENAME_FORMAT=BID24|BID64
	FTP_DEFAULT_SERVER_BCHSEP_OPT3_FILENAME_FORMAT=BID24|BID64
	FTP_DEFAULT_SERVER_DATA _PORT_RANGE=L-1 | nnnnn-nnnnn, nnnnn-nnnnn
	FTP_DEFAULT_SERVER_DIRFORM=BROWSER|BROWSER64|MBOX_CLIENT|MBOX_CLIENT64| MBOX_ZOS|MBOX_ZOS64|$MIBNSDFXY|UNIX|UNIX64
	FTP_DEFAULT_SERVER_REMOTE_FILENAME_LENGTH=SHORT|LONG|LONG64
	FTP_DEFAULT_SERVER_SCAN=NO | YES | ALL
	FTP_LOGON_REPLY=0 | nn
	FTP_LOGON_SCRIPT_DEFAULT=xxxxxxxx
	FTP_MAX_CLIENT_THREADS= 10| nnn
	FTP_MAX_SERVER_THREADS=10 | nnn
	FTP_SERVER_CONTROL_PORT=[’hostname’, | ’nnn.nnn.nnn.nnn’] nnnnn
	SCRIPT_INTERVAL_TIME=0030 | nnnn
	SSL=NO | YES
	SSL_CIPHER_SUITE=0A09050403020106 | cipher-suite-list
	SSL_DEFAULT_CLIENT_AUTH_POLICY=OPTIONAL | REQUIRED | DISALLOWED
	SSL_DEFAULT_CLIENT_CCC_POLICY=DISALLOWED | OPTIONAL | REQUIRED
	SSL_DEFAULT_POLICY=OPTIONAL | REQUIRED | DISALLOWED
	SSL_DEFAULT_SERVER_CCC_POLICY=DISALLOWED | OPTIONAL | REQUIRED
	SSL_KEY_DBASE=’key-data-base-path-name’
	SSL_KEY_DBASE_PW=’key-data-base-password’
	SSL_KEYRING_LABEL=’ring-label’
	SSL_KEYRING_NAME=’ring-name’
	SSL_SERVER_CERT=’certificate-label-string’
	SSL_TIMEOUT=00300 | nnnnn
	SYST215='your desired text &OSNAME &OSVER'

ODF Record	FTP Parameter
*REMOTES	Defines the name, type, and site-specific settings for remote FTP sites, including SSL protocol. Values defined in these records take precedence over global settings defined in the *OPTIONS record.
FTP Client	Defines the name, type, and site-specific settings for remote-initiated connections from FTP client sites to the Connect:Enterprise for z/OS repository, including generic and anonymous client definitions.
	NAME=<remote_name> | ANONYMOUS | [generic_remote]*
	TYPE=FTP_CLIENT
	BCHSEP=NONE | OPT3 | OPT4
	DIR_FILTER=D | flags
	DIRFORM=BROWSER|BROWSER64|MBOX_CLIENT| MBOX_CLIENT64| MBOX_ZOS|MBOX_ZOS64|$MIBNSDFXY|UNIX|UNIX64
	DISCINTV=0-3600
	EDI=YES|NO
	FTP_ALLOW_GETBYNBR_DFLAG=NO|YES
	FTP_DATA_PORT_RANGE=L-1 | nnnnn-nnnnn, nnnnn-nnnnn
	FTP_PORT_RETRIES=nn
	FTP_CLIENT_SYST215='your desired text &OSNAME &OSVER'
	FTP_RETRY_WAIT_TIME=nnn
	KIRN= NO | YES
	LS_FILTER=BDI!RST | flags
	ONEBATCH=NO|YES
	PASSWORD_CASE=UPPER | MIXED | BOTH
	RECEIVE_OPTIONS= (BID='NONE' | '<24 byte string>', EO=NO|YES, MULTXMIT=NO|YES, RENAME=BID | FIRST24 | LAST24 | FIRST64 | LAST64, TO=NO|YES, XMIT=NO|YES,)
	REMOTE_FILENAME_LENGTH = SHORT | LONG | LONG64
	RIFS= YES | NO
	SCAN=NO | YES | ALL
	SSL_CCC_POLICY=DISALLOWED | REQUIRED | OPTIONAL
	SSL_CLIENT_AUTH_POLICY=REQUIRED | DISALLOWED | OPTIONAL

ODF Record	FTP Parameter
	SSL_POLICY=OPTIONAL | REQUIRED | DISALLOWED
	SYST215='your desired text &OSNAME &OSVER'
	TRANSLATE=translate table name | STANDARD
FTP Server	Defines the name, type, and site-specific settings for host-initiated Auto Connect sessions to remote FTP servers.
	NAME=xxxxxxxx
	TYPE=FTP_SERVER
	&BID='NONE'|'xxx...xxx'
	&DATAMODE=B|C|S
	&IPADDR=hostname
	&NEWPASS=xxxxxx...xxx
	&PASSWORD=xxxxxx...xxx
	&PORTNO=21|nnnn
	&RECVPATH='directory_path'
	&SENDPATH='directory_path'
	&DATASTRU=F|R
	&DATATYPE=A|E|I
	&USERID=remote_name|xxxxxxxx
	BCHSEP=NONE | OPT3 | OPT4
	DISCINTV=0 |1-3600
	EDI=YES|NO
	IDENT=YES|NO
	KIRN= NO | YES
	LOGON_SCRIPT=xxxxxxxx
	FTP_DATA_PORT_RANGE=U | nnnnn-nnnnn, nnnnn-nnnnn
	FTP_CONTROL_PORT_RANGE=nnnnn-nnnnn, nnnnn-nnnnn
	FTP_PORT_RETRIES=nn
	FTP_PORT_RETRY_WAIT_TIME=nnn
	REMOTE_FILENAME_LENGTH = SHORT | LONG | LONG64
	RIFS= YES | NO
	SCAN= NO | YES | ALL

ODF Record	FTP Parameter
	SENDPASV=NO|YES
	SENDSITE=NO|YES
	SSL_CCC_POLICY=DISALLOWED | REQUIRED | OPTIONAL
	SSL_POLICY=OPTIONAL | REQUIRED | DISALLOWED
	TRANSLATE=pds_member_name| STANDARD
*CONNECT	Defines the name, type, and characteristics of Auto Connect lists.
	LISTNAME=XXXXXXXX
	TYPE=FTP
	ACQUEUE=Y | N | F
	CALENDAR=xxxxxxxx
	SESSIONS=nnn | 1
	TIME=hh:mm | [,hh:mm, ...]
FTP Remote Site Specification	Specifies the remote site or sites to contact and additional options for the sites. Values defined in these records override global FTP settings defined in the *OPTIONS record and the site-specific values set in the *REMOTES record.
	REMOTE_NAME
	AC_SCRIPT=name
	BCHSEP=NONE | OPT3 | OPT4
	ONEBATCH=NO | YES (OB=Y | N)
	&BEGINLIST=aaaaaaaa
	&IDLIST=bbbbbbbb | [,cccccccc,...]
	&ENDLIST=xxxxxxxx

ODF Records and Parameters for Configuring BSC Connections

The following table summarizes the ODF records and parameters use to configure BSC connections. Bold indicates required parameters.

ODF Record	BSC Parameter
*OPTIONS	Defines global settings for BSC connections.
	BSC_DEFAULT_$$DIR_FORMAT = BID24 | BID64

ODF Record	BSC Parameter
	BTAM=YES
	RETAIN
	RMDC=YES
	SCINCOR=YES | NO
	SECURITY=BATCH
	UA=xxxxxxxx
	WACKMAX=20 | nnn
*SECURITY	Secures batches by verifying Mailbox IDs for remote connections from BSC sites without implementing Connect:Enterprise for z/OS Security Interface. Required if SECURITY=BATCH in *OPTIONS record.
	ID=xxxxxxxx, ID=xxxxxxxx,
*SIGNON	Used to identify the valid SIGNON formats when the remote site sends a signon record to the host when a transmission connection is established.
	/*SIGNON
	$SIGNON
*IDVER	Required to define remote and host IDs to exchange when BTAM ID verification is specified for BSC lines.
	HID=xxx...xxx
	RID=xxx...xxx
*CONNECT	Defines the name, type, and characteristics of Auto Connect lists.
	LISTNAME=XXXXXXXX
	TYPE=BSCAD | BSCMD | BSCNS
	ACQUEUE=Y | N
	CALENDAR=xxxxxxxx
	DELAY=0 | nnnn
	DISCINTV=NO | nnnn | 0
	JES=NO | YES
	LINES=xxxxxxxx | [,xxxxxxxx]
	NOBATCH=C | NC
	POWER=NO | YES
	RETRY=0 | nn
	SIGNOFF=NO | YES

ODF Record	BSC Parameter
	TIME=hh:mm | [,hh:mm, ...]
BSC Remote Site Specification	Specifies the remote BSC site to contact and additional options for the site. Values defined in these records override global BSC settings defined in the *OPTIONS record.
	REMOTE_NAME
	dd
	nn...nn | Dnn...nn | CRNnn...nn
	BCHSEP=NO | OPT1 | OPT2 | OPT3
	BLOCK=nn | *nn
	CMP=N | Y
	HID=xxx...xxx
	LINEID=xxxxxxx
	MODE=SENDRECV | SENDONLY | RECVSEND | RECVONLY
	ONEBATCH=YES | NO
	RECSEP=1E | 1F
	RID=xxx...xxx
	TRANSPAR=N | Y
	TRUNC=N | Y
	BEGINLIST=xxxxxxxx
	IDLIST=xxxxxxxx
	ENDLIST=xxxxxxxx

*CALENDAR Record for Auto Connect Scheduling and Exceptions

If no *CALENDAR record is specified in the *CONNECT record, the time-initiated Auto Connect is performed daily at the times specified in the *CONNECT record. To schedule Auto Connects for activation or deactivation on specific dates or days of the week, you can create the *CALENDAR section and specify the CALENDAR= record on an Auto Connect list. The *CALENDAR record parameters are the same for all types of Auto Connects; however, each *CALENDAR record must have a unique name. See Chapter 7, *Configuring *CALENDAR Records*, for a discussion of the Auto Connect calendar processing rules.

ODF Record	Parameter
*CALENDAR	NAME=xxxxxxxx
	DATES=mm/dd
	EXDATES=mm/dd
	EXDAYS=SUN MON TUE WED THU FRI SAT

Chapter 3

Configuring *OPTIONS Record for System Resources

This chapter describes the *OPTIONS record parameters that are required to initialize the product, define its default behavior, and set default values for protocol-specific communications to and from remote sites.

Note: Connect:Enterprise for z/OS supports both the TLS (Transport Layer Security) and SSL (Secure Sockets Layer) protocols. Throughout this chapter, the phrase SSL is used to describe both the SSL and TLS protocols.

*OPTIONS Record Format and Rules

Review the *OPTIONS record format and rules for the information required to define the *OPTIONS record parameters for system resources.

*OPTIONS Record Format

The following example illustrates all the possible parameters you can configure in the *OPTIONS record; however, this chapter describes configuring only those parameters associated with the system resources as defined in the summary table in *Connect:Enterprise for z/OS System Resources* on page 27. Parameters that must be in a required position are displayed in the required order.

For information about protocol-specific parameters set in the *OPTIONS record, see Chapter 4, *Configuring ODF Records for SNA Connections*, Chapter 5, *Configuring ODF Records for FTP Connections*, and Chapter 6, *Configuring ODF Records for BSC Connections*.

```
*OPTIONS
** THIS IS A COMMENT CARD, COMMENTS START WITH "**".
   ACQDEFAULT=YES|NO
   APDSN=YOUR.ASSET.PROTECT.FILE.DSNAME
   APPC=YES|NO
   APPCAPPL=XXXXXXXX
   APPCPLSZ=316
   APPLID=ENTPRS
   BROWSE_AUTOCLEAN_INTERVAL=60|NNNNN
   BROWSE_DATASPACE_COUNT_MAX=20|NNN
   BROWSE_DATASPACE_SIZE_MAX=2000|NNNNNN
   BROWSE_SESSION_COUNT_MAX=40|NNNN
   BROWSE_SESSION_RETIREMENT_AGE=300|NNNNN
   BSC_DEFAULT_$$DIR_FORMAT=BID24|BID64
   BTAM=YES|NO
   CICSAPPL=XXXXXXXX
   CICSMODE=XXXXXXXX
   CICSTR1=xxxx | CM62
   CMB001I='ENTER C:E REQUEST WHEN READY'
   CONSLOG=YES|NO
   CONSOLEDESC=NN
   CONSOLEROUT=NN
   COUNT=BLOCK|RECORD
   CSC_DEFAULT_REPORTS_FORMAT=1|1X|2
   DALLOC_RETRY_INTERVAL=30|NNNN
   DALLOC_VBQ_INUSE=FAIL|RETRY
   DALLOC_VBQ_STOUTL=ALLOW|DISALLOW
   DALLOC_VLF_INUSE=FAIL|RETRY
   DALLOC_VLF_STOUTL=ALLOW|DISALLOW
   DEFAULT_MODE=BID24|BID64
   DIALOG_FTP=*
   FTP=NO|YES
   FTP_AC_SCRIPT_DEFAULT=LEVI
   FTP_ALLOW_GETBYNBR_DFLAG_DEFAULT=NO|YES
   FTP_CONNECT_INTERVAL=0060|NNNN
   FTP_CLIENT_PASV_DATA_IPADDR=R227|CPADDR
   FTP_DEFAULT_CLIENT_BCHSEP_NONE_FILENAME_FORMAT=BID24|BID64
   FTP_DEFAULT_CLIENT_BCHSEP_OPT3_FILENAME_FORMAT=BID24|BID64
   FTP_DEFAULT_CLIENT_CONTROL_PORT_RANGE=NNNNN-NNNNN,NNNNN-NNNNN
   FTP_DEFAULT_CLIENT_DATA_PORT_RANGE=U|NNNNN-NNNNN,NNNNN-NNNNN
   FTP_DEFAULT_CLIENT_LOCDIRFORM=BROWSER|BROWSER64|MBOX_CLIENT|MBOX_CLIENT64|
     MBOX_ZOS|MBOX_ZOS64|$MIBNSDFXY|UNIX|UNIX64
   FTP_DEFAULT_CLIENT_REMOTE_FILENAME_LENGTH=SHORT|LONG|LONG64
   FTP_DEFAULT_CLIENT_SCAN=NO|YES|ALL
   FTP_DEFAULT_DIALOG_TRACE_LRECL=136 | nnnnn
   FTP_DEFAULT_DISCINTV=0900|NNNN
   FTP_DEFAULT_KIRN=YES | NO
   FTP_DEFAULT_PORT_RETRIES=NN|0
   FTP_DEFAULT_PORT_RETRY_WAIT_TIME=NNN|030
   FTP_DEFAULT_RECEIVE_OPTION_RENAME=FIRST24|LAST24|FIRST64|LAST64
   FTP_DEFAULT_RIFS=YES | NO
   FTP_DEFAULT_SERVER_BCHSEP_NONE_FILENAME_FORMAT=BID24|BID64
   FTP_DEFAULT_SERVER_BCHSEP_OPT3_FILENAME_FORMAT=BID24|BID64
```

Continued

```
   FTP_DEFAULT_SERVER_DATA _PORT_RANGE=L-1|NNNNN-NNNNN,NNNNN-NNNNN
   FTP_DEFAULT_SERVER_DIRFORM=BROWSER|BROWSER64|MBOX_CLIENT|MBOX_CLIENT64|
      MBOX_ZOS|MBOX_ZOS64|$MIBNSDFXY|UNIX|UNIX64
   FTP_DEFAULT_SERVER_REMOTE_FILENAME_LENGTH=SHORT|LONG|LONG64
   FTP_DEFAULT_SERVER_SCAN=NO|YES|ALL
   FTP_LOGON_SCRIPT_DEFAULT=LGNLEVI
   FTP_MAX_CLIENT_THREADS=10|NNNN
   FTP_MAX_SERVER_THREADS=10|NNNN
   FTP_SERVER_CONTROL_PORT=5555|NNNN
   FTP_LOGON_REPLY=1
YOU ARE LOGGED ONTO C:E SECURE FTP SERVER XXXXXXXX.
   ICO_DEFAULT_REPORTS_FORMAT=1|1X|2
   LOGONMSG='SUCCESSFUL LOGON TO Connect:Enterprise'
   MAXCP=X
   MAXRP=X
   MAXRWAIT=HH:MM:SS,C
   MBXHLQ=XXXXXXXX
   MBXNAME=XXXXXXXX
  MAXCP=nn | 2
  MAXRP=nn | 2
  MAXRWAIT=HH:MM:SS,C
  MBXHLQ=xxxxxxxx | MAILBOX
  MBXNAME=xxxxxxxx | MAILBOX
  MBXSECURE=LOGON | BATCH | ALL | WARN | OFF
  APISECURE=LOGON|BATCH|WARN|ALL|OFF
  BSCSECURE=LOGON|BATCH|WARN|ALL|OFF
  CSCSECURE=LOGON|BATCH|WARN|ALL|OFF
  FTPSECURE=LOGON|BATCH|WARN|ALL|OFF
  ICOSECURE=LOGON|BATCH|WARN|ALL|OFF
  SNASECURE=LOGON|BATCH|WARN|ALL|OFF
  STLSECURE=BATCH|WARN|OFF
  UIFSECURE=LOGON|BATCH|WARN|ALL|OFF
  MODIFY=YES | NO | RESP
  PASSWORD=xxxxxxxx
  PASSWORD_CASE=UPPER | MIXED | BOTH
  RETAIN
  RMDC=YES
  RULES=YES | NO
  RULES_IR=YES | NO
  RULES_RECURSION_MAX=5 | 0-2147483647
  RULES_CN=YES | NO
  RULES_CN_PREFIX=xx
  RULESCON=xxxxxxxx
  RULESEOB=xxxxxxxx
  RULESLOG=xxxxxxxx
  RULESSCH=xxxxxxxx
  RULESWKT=xxxxxxxx
  SCINCOR=YES | NO
  SCRIPT_INTERVAL_TIME=0030 | nnnn
  SECURITY=LOGON
  SECURITY=BATCH
  SNA_DEFAULT_$$DIR_FORMAT=BID24|BID64
```

Continued

```
  SSL=NO | YES
  SSL_CIPHER_SUITE=0A09050403020106 | cipher-suite-list
***********************************************************************
**         KEY DATA BASE         VERSES     KEYRING                  **
***********************************************************************
      SSL_KEY_DBASE_PW='SEC'
      SSL_KEY_DBASE='/U/MYDATA/'   OR   SSL_KEYRING_LABEL='RING-LABEL'
      SSL_SERVER_CERT='CERT-LABEL'      SSL_KEYRING_NAME='RING-NAME'
***********************************************************************
  SSL_TIMEOUT=00300|NNNNN
  SSL_DEFAULT_CLIENT_AUTH_POLICY=OPTIONAL | REQUIRED | DISALLOWED
  SSL_DEFAULT_CLIENT_CCC_POLICY=DISALLOWED | OPTIONAL | REQUIRED
  SSL_DEFAULT_SERVER_CCC_POLICY=DISALLOWED | OPTIONAL | REQUIRED
  SSL_DEFAULT_POLICY=OPTIONAL | REQUIRED | DISALLOWED
  STOUTL_DEFAULT_REPORTS_FORMAT=1|1X|2
  SUMMARY=ONLY | FINAL | ANY
  SYSOUTCLASS=X | x
  SYST215='your desired text &OSNAME &OSVER'
  TCPSCH=xxxxxxx
  TRACE=ALLTP|APO|APQ|CP|EXITS|PR|SNA|TCPSCH|VA2C|VSAM
  TRACE=RPEOB|RPLOG|RPWKT|RPSCH|RPCON
  TRACE_FTP=*
  TRACEID=xxxxxxxx
  UA=ANYNAME          <===== USER ASSEMBLY SYSLMOD NAME
  VBQPCT=NN
  VBQROTAT=NN
  VLFPCT=NN
  VLFROTAT=NN
  VPF='YOUR.ENTPRS.VPF.DSNAME'
  VSESSLIM=NN
  VTAM=YES
  WACKMAX=NN
  XAPPCSEC=XXXXXXXX
  XAPPCCWI=XXXXXXXX
  XAPPCCWT=XXXXXXXX
  XENDOFB=XXXXXXXX
  XEOBVER=2
  XINIT=XXXXXXXX
  XINPUT=XXXXXXXX
  XLOG=XXXXXXXX
  XOUTPUT=XXXXXXXX
  X_SECURE=STSECFTP
  XSECUR1=XXXXXXXX
  XSECUR2=XXXXXXXX
  XTERM=XXXXXXX
**
```

*OPTIONS Record Rules

When you define the *OPTIONS parameters, observe the following rules:

- Create the *OPTIONS record as the first record of the ODF.
- Type *OPTIONS in column 1of line 1 of the ODF. Any other text on line 1 is ignored.
- The following *OPTIONS parameters are required to initialize the product:
  - APDSN=data.set.name
  - DEFAULT_MODE=BID24|BID64
  - VPF=’data.set.name

*OPTIONS Record Parameters for System Resources

The following table lists the *OPTIONS record parameter definitions. Required parameters are listed in bold in the required positional in the table; the remaining optional parameters are listed alphabetically.

Parameter	Description
APDSN=data.set.name	Required to initialize the product. Specifies the required 1–44 byte z/OS data set name that contains the AP Key file.
DEFAULT_MODE=BID24|BID64	Required to initialize the product. Specifies the default value for a subset of 15 ODF parameters which determine the format Connect:Enterprise uses for the user batch ID (BID) in displays, reports, and traces. ◆ BID24—Connect:Enterprise sets the defaults for a 24 character User Batch ID. ◆ BID64—Connect:Enterprise sets the defaults for a 64 character User Batch ID. You can override the defaults for individual parameters that this parameter sets. For more details, see the *Connect:Enterprise for z/OS Release Notes.*
VPF=‘data.set.name’	Required to initialize the product. Specifies the data set name of the VSAM Pointer File. The VPF provides the definition for the files defined to Connect:Enterprise. The name specified on the VPF control record must match the name used in the offline PURGE utility VPF control record, which was used when Connect:Enterprise batch files were initialized.
ACQDEFAULT=YES | NO	(ACQDEF) Used for Auto Connect sessions. Specifies the default value used by the ACQUEUE parameter in the *CONNECT options. If ACQUEUE is not specified for a specific Auto Connect list name, the ACQDEFAULT is used.

Parameter	Description
APISECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for APPC LU6.2 connections only. This category of transactions includes requests that do not originate from a Connect:Enterprise for z/OS product component, such as ICO or the ISPF user interface. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for API (LU6.2) connections, while allowing the security interface to remain active for other protocols.
APPC=YES | NO	If FTP=YES, APPC=YES is required. Specifies that the APPC interface is started. YES is required for CICS, ISPF, and ICO interfaces.
APPCAPPL=xxxxxxxx	The ACB name in VTAM opened by the APPC interface for use with CICS and ISPF interfaces. This parameter is required if APPC=YES is specified.
APPCPLSZ=nnnn | 316	Specifies the number (range 64–9999) of 4-KB pages allocated to the APPC storage pool if APPC=YES is specified. InterCONNECT Option (ICO) or Connect:Enterprise CICS API interface users may want to specify the initial APPCPLSZ as 316 to be consistent with the original Connect:Enterprise design setting supporting concurrent storage demand. To assist tuning the APPCPLSZ parameter, message CMB289I is issued at Connect:Enterprise shutdown to show the allocated and used pages consumed during that Connect:Enterprise execution.

Parameter	Description
BROWSE_AUTOCLEAN_INTERVAL= 60 | nnnn	The maximum number of seconds between automatic cleanup cycles. Valid values range from 0 to 32767. The default value is 60. The cleanup cycle deletes any browse data space that has been unused for the number of seconds specified in BROWSE_SESSION_RETIREMENT_AGE. A regular (synchronous) cleanup cycle occurs every time any batch is browsed. An automatic (asynchronous) cleanup cycle occurs when the time set in BROWSE_AUTOCLEAN_INTERVAL elapses after either type of cleanup. If BROWSE_SESSION_RETIREMENT_AGE is set to 0, the autoclean interval value is ignored and neither type of cleanup is performed. If BROWSE_SESSION_RETIREMENT_AGE is set to a value other than 0, and BROWSE_AUTOCLEAN_INTERVAL is set to 0, only regular cleanups occur. If values other than zero are set for both BROWSE_AUTOCLEAN_INTERVAL and BROWSE_SESSION_RETIREMENT_AGE, both types of cleanup cycles are performed.
BROWSE_DATASPACE_COUNT_MAX =20 | NNN	The maximum number of concurrent browse data spaces allowed. Valid values range from 0 to 480. If the value is set to 0, no browse data spaces are created, and the browse online interfaces (CICS and ISPF) function as they did before Connect:Enterprise, versions 1.1.00 and earlier. If the creation of a browse data space exceeds the limit set in this value, the space which has been unused for the longest time is deleted and the new data space is created.
BROWSE_DATASPACE_SIZE_MAX= 524288 | nnnnnn	The maximum number of pages of storage allotted to any one data space. Valid values range from 1 to 524288 (approximately 2 GB of space). If the batch being loaded into the browse data space exceeds this value, the browse terminates with error code 0600 and the browse data space is deleted. Data space virtual storage is handled the same as regular address space virtual storage. Therefore, specifying a high value in this parameter does not cause large storage consumption but it does enable it.

Parameter	Description
BROWSE_SESSION_COUNT_MAX= 40 | nnnn	Sets the maximum number of concurrent sessions allowed. Valid values range from 0 to 1023. BROWSE_SESSION_COUNT_MAX must be at least as large as BROWSE_DATASPACE_COUNT_MAX. A session associates a user with a browse data space. Sessions are only deleted by cleanup cycles. If the deleted session was the only one associated with its browse data space, the data space is deleted. Thus a low ratio of BROWSE_SESSION_COUNT_MAX to BROWSE_DATASPACE_COUNT_MAX can cause browse data spaces to be deleted before BROWSE_SESSION_RETIREMENT_AGE has been reached.
BROWSE_SESSION_ RETIREMENT_AGE=300 | nnnn	Sets the number of seconds a browse data space is protected from being deleted by a cleanup cycle. Valid values range from 0 to 32767. The default is 5 minutes (300 seconds). If the value set in BROWSE_SESSION_RETIREMENT_AGE is 0, BROWSE_AUTOCLEAN_INTERVAL is ignored and no cleanup cycle occurs.
BSCSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for bisync connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for bisync connections, while allowing the security interface to remain active for other protocols.
CICSAPPL=xxxxxxxx	The CICS ACB name. This is the LU name Connect:Enterprise uses to initiate a conversation with CICS. This parameter is required for CICS and when APPC=YES is specified.
CICSMODE=xxxxxxxx	The Mode Entry name to use when initiating a conversation with CICS. This parameter is required for CICS and when APPC=YES is specified.

Parameter	Description
CICSTR1=xxxx | CM62	The Connect:Enterprise CICS interface LU6.2 transaction name. This may have been altered from the default during CICS application installation. This parameter is required for CICS and when APPC=YES is specified.
CMB001I='xxx...xxx'	Supplies your own version of the Connect:Enterprise prompt message that is displayed on the host system console while Connect:Enterprise is executing. If this parameter is omitted, the standard prompt is displayed: CMB001I–ENTER Connect:Enterprise REQUEST WHEN READY If this parameter is supplied, the 1–60 character message is enclosed in quotes, with no embedded quotes.
CONSLOG=YES | NO	Places a write-to-operator (WTO) message containing the remote name on the host site console whenever a Connect:Enterprise session starts or ends. ◆ YES—Connect:Enterprise activates Console Logging for remote-initiated connections and disconnections. ◆ NO—Connect:Enterprise does not put the WTO message on the host site console.
CONSOLEDESC=nn | 07	Specifies the z/OS Console Message Descriptor Code used for all Connect:Enterprise console messages. Descriptor codes classify console messages into certain defined types. The descriptor code values are defined in the WTO macro in the IBM manuals. The code is 2 digits, value 01–16. The default value (07) deletes any action messages at the end of the job or task.
CONSOLEROUT=nn | 01	Specifies the 2-digit z/OS Console Routing Code for all Connect:Enterprise console messages. Used to display Connect:Enterprise console messages on consoles defined for special routing codes during your z/OS sysgen process. The routing code values are defined in the WTO and WTOR macros in the IBM manuals. Valid values are 01–16. The default value (01) displays all Connect:Enterprise messages on the Master Console.
COUNT=RECORD | BLOCK	Indicates how count information is presented to console operators and remote users when using the $$DIR command. ◆ RECORD—Displays the logical record count. ◆ BLOCK—Displays the number of physical blocks received.

Parameter	Description
CSC_DEFAULT_REPORTS_FORMAT= 1 | 1X | 2	Specifies the default reports format for the CSC (Cross System Client) SYSPRINT and REPORTS DD file. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, 1 is the default for this parameter; if BID64 is specified, 1X is used for this parameter. If specified, this value is used for all CSC reports for which there is no explicit FORMAT= parameter coded in any given CSC SYSIN command, such as, ADD or STATFLG. ◆ 1—Prints the normal (original) report's single detail line items, which display only 24 characters of the User Batch ID. ◆ 1X—Prints single line extended detail items to accommodate the full 64 character User Batch ID. ◆ 2—Prints two lines for each detail item. The first detail line is formatted using format 1 (i.e., the original format with the 24 character User Batch ID). The second detail line item prints only the fully qualified 64 character User Batch ID, aligned with the 24 character Batch ID on line one above.
CSCSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for Cross System Client (CSC) APPC LU6.2 connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for CSC (LU6.2) connections, while allowing the security interface to remain active for other protocols.
DALLOC_RETRY_INTERVAL= <u>30</u>–-3600	Specifies the number of seconds the server waits between attempts to deallocate files. If a $$DALLOC command cannot be processed and the request can be retried, the command is queued. Each time this interval expires, all queued $$DALLOC commands are processed. If the file is still in use, the command is requeued until the deallocation is successful.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>DALLOC_VBQ_INUSE=FAIL | RETRY</td><td>Specifies the default behavior for all $$DALLOC VBQnn commands in which the INUSE=FAIL/RETRY parameter is omitted.
◆ FAIL—Fails the request if the file is currently in use and unable to be immediately deallocated.
◆ RETRY—Retries the command later if the file is currently in use. The request is queued, then reissued at each retry interval until successful.</td></tr>
<tr><td>DALLOC_VBQ_STOUTL=
ALLOW | DISALLOW</td><td>Specifies the default behavior for all $$DALLOC VBQnn commands in which the STOUTL=ALLOW/DISALLOW parameter is omitted.
◆ ALLOW—Allows STOUTL to access the deallocated VBQ.
◆ DISALLOW—Does not allow STOUTL to access the deallocated VBQ.</td></tr>
<tr><td>DALLOC_VLF_INUSE=FAIL | RETRY</td><td>Specifies the default behavior for all $$DALLOC VLF commands in which the INUSE=FAIL/RETRY parameter is omitted.
◆ FAIL—Fails the request if the file is currently in use and unable to be immediately deallocated.
◆ RETRY—Retries the command later if the file is currently in use. The request is queued, then reissued at each retry interval until successful.</td></tr>
<tr><td>DALLOC_VLF_STOUTL=ALLOW | DISALLOW</td><td>Specifies the default behavior for all $$DALLOC VLFn commands in which the STOUTL=ALLOW/DISALLOW parameter is omitted.
◆ ALLOW—Allows STOUTL to access the deallocated VLF.
◆ DISALLOW—Does not allow STOUTL to access the deallocated VLF.</td></tr>
<tr><td>DIALOG_FTP=NO|*|
remote1|remote1*,.....|
remote8|remote8*</td><td>Activates dialog session tracing for FTP remote sites. Remote definitions can be explicit or generic. To capture the dialog flow for FTP Client Auto Connect sessions, specify'DIALOG_FTP=*'.
To capture the dialog for selected remote sites, specify:
remote1,remote2,remote3,.....,remote8.
Remote names can refer to either an FTP_CLIENT or an FTP_SERVER.
Dialogs are not stored as part of the batch. The data received and sent as a part of the batch is not a part of the dialog. Dialogs are continuously written to the Connect:Enterprise FTP Logging Address Space. Dialogs are not limited to one function type if two remote definitions have the same name.</td></tr>
</table>

Parameter	Description
FTPSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for FTP connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for FTP connections, while allowing the security interface to remain active for other protocols.
ICO_DEFAULT_REPORTS_ FORMAT=1 | 1X | 2	Specifies the default reports format for the ICO (Inter-Connect Option) SYSPRINT and REPORTS DD file. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, 1 is the default for this parameter; if BID64 is specified, 1X is used for this parameter. If specified, this value is used for all ICO reports for which there is no explicit FORMAT= parameter coded in any given CSC SYSIN command, such as, ADD or EXTRACT. ◆ 1—Prints the normal (original) report's single detail line items, which display only 24 characters of the User Batch ID. ◆ 1X —Prints single line extended detail items to accommodate the full 64 character User Batch ID. ◆ 2—Prints two lines for each detail item. The first detail line is formatted using format 1 (i.e., the original format with the 24 character User Batch ID). The second detail line item prints only the fully qualified 64 character User Batch ID, aligned with the 24 character Batch ID on line one above.

Parameter	Description
ICOSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for InterConnect (ICO) APPC LU6.2 connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for ICO (LU6.2) connections, while allowing the security interface to remain active for other protocols.
MAXCP=nn | 2	Specifies the maximum number of command processor tasks (from 1–99) allowed to run concurrently. This parameter is limited by the storage available in your system. The tasks parallel schedule units of work for APPC. This parameter does not control the number of concurrent ISPF or CICS interface users, only how fast their requests are serviced. To assist tuning the MAXCP parameter, message CMB297 is issued at Connect:Enterprise shutdown.
MAXRP=nn | 2	Specifies the maximum number of rules processor tasks (1–99) allowed to run concurrently for application agent processing.
MBXHLQ=xxxxxxxx | MAILBOX	1–8 character string used as the high-level qualifier for creating a pseudo data set name. This value is used to check batch function security when the security interface is active.
MBXNAME=xxxxxxxx | MAILBOX	Specifies the 8-character mailbox name. This name is used by the Security interface when making BATCH/FUNCTION security checks. It is also used by the application agents when dynamically defining consoles to issue console commands.

Parameter	Description
MBXSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the global Connect:Enterprise security interface. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. If you do not specify a value for the MBXSECURE parameter, the security interface does not do any security checking. **Note:** This parameter controls the security interface at a system-wide level and can be selectively overridden by other protocol/component-specific ODF security parameters, including APISECURE, BSCSECURE, CSCSECURE, FTPSECURE, ICOSECURE, SNASECURE, STLSECURE, and UIFSECURE. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking at the global level, that is, for all protocols. This is the default value. You cannot change the MBXSECURE parameter by using the ISPF or CICS interfaces.
MODIFY=YES | NO | RESP	Enables Connect:Enterprise to use the z/OS modify command interface for entering Connect:Enterprise $$ commands. ◆ YES—Uses the z/OS systems MODIFY interface to type Connect:Enterprise commands and return the responses to the CONSOLEROUT specification. This requires Connect:Enterprise to be a console-started task. ◆ NO—Uses WTOR to type Connect:Enterprise commands. When Connect:Enterprise uses WTOR to issue messages that require a response, an action character precedes the text of the message. The actual characters used for the action character varies, depending on if Connect:Enterprise is executing in an APF-authorized environment. ◆ RESP—Uses the z/OS systems MODIFY interface to type Connect:Enterprise commands and return the responses only to the console that entered the commands. This requires Connect:Enterprise to be a console-started task.
PASSWORD=xxxxxxxx	A 1–8 character string specifying the system password for accessing restricted Connect:Enterprise functions such as full directory listings. This parameter is required and enables remote sites to list the entire VSAM batch files directory. The PASSWORD is printed as eight question marks on the SYSPRINT listing.

Parameter	Description
PASSWORD_CASE=UPPER | MIXED | BOTH	Specifies how passwords are presented to the security package at logon authorization, in terms of case sensitivity. ◆ UPPER—The password is uppercased before it is presented to the security package for logon security authorization. Only the upper cased value is validated by the security package. For example, if the user enters a password value of "MyPass", the value is uppercased to "MYPASS". If a new password is provided in the logon request, it is also uppercased. ◆ MIXED—The password is not uppercased before it is presented to the security package for logon security authorization. Only the original password value, as entered by the user, is validated by the security package. ◆ BOTH—Both mixed and uppercase password values are validated by the security package, if necessary. First, the original value (mixed case) is validated by the security package. If the logon check fails using the mixed case password, a second attempt is made to validate the password using the uppercased value. A successful logon check with either mixed or upper cased password is considered valid. The intended purpose of specifying BOTH is to allow a transition period for some duration after your systems programmer / security administrator turns on mixed case password support in the security package. This enables Connect:Enterprise to successfully allow users/remotes to logon. Eventually, after users have changed the password with mixed case turned on in the security package, you should set PASSWORD_CASE=MIXED. **Note:** When BOTH is specified, if the first attempt fails (mixed case), but the second attempt is successful (uppercase), Connect:Enterprise considers the logon successful and continues processing as normal. However, the security package still posts a security violation console message due to the failure of the first logon check attempt in mixed case. Also be aware that if both attempts fail, the "consecutive unsuccessful password attempts" count, maintained within the security package, is incremented by 2, since two individual calls are made to the security package.
RULES=YES | NO	Activates application agent processing. RULESEOB, RULESLOG, or RULESWKT are also specified. An application agent is an interface which enables you to customize the Connect:Enterprise environment. The agents are fully described in the *Connect:Enterprise for z/OS Application Agents and User Exits Guide*.

Parameter	Description
RULES_CN=YES | NO	Specifies whether or not a dynamic (unique) console name (CN) is generated each time a rules COMMAND instruction is processed. YES = The console name generated is dynamic for each rules COMMAND instruction processed. The console name is an 8-character value in format xxnnssss. xx = A user specified console name prefix. The prefix is set by specifying the RULES_CN_PREFIX=xx parameter. A two character value must be specified. The default prefix is "RP" (Rules Processor). nn = The Rules Processor subtask number (01-99) processing this COMMAND instruction. ssss = A sequence number (0001-9999) that is incremented each time a COMMAND instruction is processed. When the sequence number reaches 9999, it is reset and starts over at 0001. Each Rules Processor subtask maintains its own sequence number. NO = A static console name is used for each rules COMMAND instruction processed. The console name assigned is equal to the value specified in the ODF *OPTIONS MBXNAME=xxxxxxxx parameter. If MBXNAME= is not specified in the ODF, the default value of "MAILBOX" is used as the console name.
RULES_CN_PREFIX=xx	Specifies a two-character console name prefix to be used each time a rules COMMAND instruction is processed. This value is in effect only if RULES_CN=YES is also specified, otherwise this parameter is ignored. If RULES_CN=YES is specified, but RULES_CN_PREFIX=xx is not, the default prefix is "RP" (for Rules Processor).

Parameter	Description
RULES_IR=YES | NO	Requires RULES=YES. Determines if an internal reader is dynamically chosen for each RP task. ◆ YES—Attempts to dynamically allocate an internal reader for each RP task to ddname IRRP00*nn*, where *nn* is the RP task ID number (1-99). The dynamic allocation occurs the first time the RP task processes a SUBMIT statement. If the dynamic allocation or open fails, Connect:Enterprise falls back to using the JESRDR allocation specified in the JCL. Fallback occurs on a task by task basis, such that each RP task is independent of the others. ◆ NO—Uses the internal reader the RP task used the first time it processed a SUBMIT statement for the life of the Connect:Enterprise main address space. If an RP task ABENDs, any dynamically allocated internal reader DCB is closed, but the DD remains allocated. If ESTAE=YES is in effect for the Connect:Enterprise main task, Connect:Enterprise reattaches the RP task and the next time that RP task processes a SUBMIT statement, it continues using the DCB it used before the ABEND automatically reopening a dynamically allocated internal reader.
RULES_RECURSION_MAX= nnnnnnnnnn | 5	Defines number of times console application agent-to-console application agent recursions (C2C recursions) are allowed to loop. 5 = Default; allows some recursion while preventing a loop from overwhelming the job or system log with messages. 0 = No C2C recursion is allowed. nnnnnnnnnn = 2147483647. For practical purposes, sets no limit.
RULESCON=xxxxxxxx	Requires RULES=YES. A 1–8 character name specifying the console rules member in the RULES data set. The console rules member specified is invoked whenever a match occurs on all console message criteria specified.
RULESEOB=xxxxxxxx	Requires RULES=YES. A 1–8 character name specifying the end of batch rules member in the RULES data set. The end of batch rules member specified is invoked whenever an online batch collection is completed.
RULESLOG=xxxxxxxx	Requires RULES=YES. A 1–8 character name specifying the logging rules member in the RULES data set. The logging rules member is invoked whenever Connect:Enterprise writes a log record for Auto Connect sessions, Remote Connect sessions, or Queued Auto Connect sessions.
RULESSCH=xxxxxxxx	Requires RULES=YES. A 1–8 character name specifying the scheduler rules member in the RULES data set. The scheduler rules member is invoked whenever a match occurs on all scheduling criteria specified.

Parameter	Description
RULESWKT=xxxxxxxx	Requires RULES=YES. A 1–8 character name specifying the wake-up terminate rules member in the RULES data set. The wake up terminate rules member is invoked whenever CICS sends a wake up terminate message to Connect:Enterprise, acknowledging a prior wake up initiate sent from Connect:Enterprise to CICS.
SNASECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for SNA connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for SNA connections, while allowing the security interface to remain active for other protocols.
STLSECURE=BATCH | WARN | OFF	Activates the Connect:Enterprise security interface for STOUTL offline utility functions only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. **Note:** STLSECURE applies to all executions of STOUTL, including the original version of InterConnect, that is, when activated from Connect:Direct via DMSTOUTL. ◆ BATCH—Activates batch function checking only. ◆ WARN—Activates batch function checking without causing security requests to fail after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for STOUTL offline utility functions, while allowing the security interface to remain active for other protocols.

Parameter	Description
STOUTL_DEFAULT_REPORTS_ FORMAT=1 | 1X | 2	Specifies the default reports format for the STOUTL REPORTS DD file. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, 1 is the default for this parameter; if BID64 is specified, 1X is used for this parameter. If specified, this value is used for all STOUTL reports for which there is no explicit FORMAT= parameter coded in any given STOUTL SYSIN command, such as, ADD or DELETE. ◆ 1—Prints the normal (original) report's single detail line items, which display only 24 characters of the User Batch ID. ◆ 1X—Prints single line extended detail items to accommodate the full 64 character User Batch ID. ◆ 2—Prints two lines for each detail item. The first detail line is formatted using format 1 (i.e., the original format with the 24 character User Batch ID). The second detail line item prints only the fully qualified 64 character User Batch ID, aligned with the 24 character Batch ID on line one above.
SUMMARY=ONLY|ANY|FINAL	Specifies how you want to record failure codes for Auto Connects that must be retried. Certain failure codes, which report failures at the Auto Connect level, are automatically written to the Summary log record and are not affected by the SUMMARY parameter. The following failure codes are written to the Summary record regardless if this parameter is used: 02-07, 09, 10, 12, 20, 24, 25, 40, 70, 74 and 78. ◆ ONLY—Failure codes are kept where they are written, either the ACSUMMARY or ACDETAIL record, and are not propagated. ◆ ANY—The first failure code is propagated. Use this value to record any failure code regardless of whether the final retry was successful. ◆ FINAL—The first failure code is propagated if after the final retry the failure code is nonzero. **Note:** The FINAL option applies only to SNA/BSC Auto Connects because the RETRY feature is not available for Remote Connects and FTP Auto Connects.
SYSOUTCLASS=X | x	Used by the SYSOUT file for FTP session dialog tracing.

Parameter	Description
SYST215='your desired text &OSNAME &OSVER'	Specifies the FTP server SYST 215 reply text for all FTP servers. To substitute the operating system name and version, use the &OSNAME and &OSVER variables. The default is: 215 MVS OSNAME OSVER is the operating system for Connect:Enterprise Vxx.Rxx.Mxx **Note:** To set the FTP Server SYST 215 reply text for a particular FTP server, add SYST215='your desired text &OSNAME &OSVER' to your ODF *REMOTE section. For more information, see page 143.
TCPSCH=xxxxxxx	Specifies the single remote for whose TCP scheduler activity you want to monitor. **Note:** TRACE=TCPSCH must be active to use this feature.
TRACE=ALLTP | APO | APQ | CP | EXITS | PR | SNA | TCPSCH | VA2C | VSAM	Invokes the Connect:Enterprise trace capabilities for debugging purposes. It captures snapshot dumps of Connect:Enterprise control information that can trace line activity if problems are detected during communication with a remote site. TRACE also helps debug user-written exit programs. **Note:** For all traces involving application agents, refer to the *Connect:Enterprise for z/OS Application Agents and User Exits Guide* for more information. ◆ ALLTP—Traces all teleprocessing activity on all Connect:Enterprise sessions. ◆ APO—Traces the APPC interface for all APPC macro completions. ◆ APQ—Traces all APPC interface activity to and from the Process router. ◆ CP—Traces activity to and from command processes. ◆ EXITS—Traces information passed between user-supplied exit programs before and after each CALL to an exit. ◆ PR—Traces activity to and from the Process Router. ◆ SNA—Traces VTAM exit activity on all Connect:Enterprise sessions. Every LOGON attempt is traced. Additionally, unusual SNA commands, LOGON rejects, and other unique conditions are traced. Use this option to test a new Connect:Enterprise system. ◆ TCPSCH—Activates tracing for the TCP Scheduler. ◆ VA2C—Traces all requests to Connect:Enterprise file servers. ◆ VSAM—Traces all accesses to the VSAM batch files.

Parameter	Description
TRACE=RPCON | RPEOB | RPLOG | RPSCH | RPWKT	Invokes the Connect:Enterprise trace capabilities for debugging purposes. It captures snapshot dumps of Connect:Enterprise control information that can trace line activity if problems are detected during communication with a remote site. TRACE also helps debug user-written exit programs. **Note:** For all traces involving application agents, refer to the *Connect:Enterprise for z/OS Application Agents and User Exits Guide* for more information. ◆ RPCON—Traces all activity processing for all console application agent requests. ◆ RPEOB—Traces all activity processing for all end of batch application agent requests. ◆ RPLOG—Traces all activity processing for all Logging application agent requests. ◆ RPSCH—Traces all activity processing for all scheduler application agent requests. ◆ RPWKT—Traces all activity processing for all Wake-Up Terminate application agent requests.
TRACE_FTP=NO|*|remote1|remote1*|...|remote8|remote8*	Activates file transfer tracing for specific, general or all remote server sessions. If DIALOG_FTP tracing is specified for a remote site, the same dialog/trace file is used to record tracing the data transfer.
TRACEID=xxxxxxxx	A debugging tool that specifies the use of the trace facilities for a single session by its remote name or line ID. For SNA, the remote name is defined in the *REMOTES section. For BSC, the line ID is specified on M$LINEX in the user assembly. **Note:** A large amount of trace output is produced when using this parameter. Only use TRACEID= if requested by Sterling Commerce Customer Support.

Parameter	Description
UIFSECURE=LOGON | BATCH | ALL | WARN | OFF	Activates the Connect:Enterprise security interface for the CICS and ISPF User Interface APPC LU6.2 connections only. If specified, this parameter value overrides the global ODF security parameter, MBXSECURE. If not specified, this parameter inherits the same value in effect for MBXSECURE. See more information about security in the *Connect:Enterprise for z/OS User's Guide* before setting this parameter. ◆ LOGON—Activates logon checking only. ◆ BATCH—Activates batch function checking only. ◆ ALL—Activates both logon and batch function checking. ◆ WARN—Activates both logon and batch function checking without causing security requests to fail after a violation. WARN causes an error message to display after a violation. You can use this option to phase in the security interface without stopping any processing. ◆ OFF—Deactivates the security interface checking. Use this option to selectively exclude security checking for ISPF/CICS User Interface (LU6.2) connections, while allowing the security interface to remain active for other protocols.
VBQPCT=50 | nn	Specifies a maximum capacity for the current collection VBQ file. The system automatically switches to a new collection file. Specify the percentage from 50–99 of the VBQ file. ◆ nn—For example, VBQPCT=90 permits the current collection file to reach 90% of capacity before the system switches to the next VBQ. When this percentage is reached or the file has obtained a new extent, Connect:Enterprise automatically switches to the next eligible (currently allocated) VBQ collection file within the VBQROTAT range.
VBQROTAT=1|nn	Specifies the number of VBQ files (from 1 to 20) that are eligible for automatic collection. For example, specifying a value of 5 places the first five VBQ files into the rotation scheme. When VBQ05 fills up to the value specified in VBQPCT, the collection file is rotated back to VBQ01. Connect:Enterprise places the next collection in VBQ01. If no suitable rotate file is found, the collection file does not change. All collections in progress are finished in the same collection file in which they were started. Only new collections are switched to the new collection file. If this option is not specified or 1 is specified, no automated VBQ rotation occurs. The default is 1.

Parameter	Description
VLFPCT=50 | nn	Specifies a maximum capacity for the current VLF log file. The system automatically switches to a new log file. Specify the percentage from 50–99 of the VLF file. nn—A value of 90 permits the current log file to reach 90% of capacity before the system switches to the next VLF. When this percentage is reached or the file has obtained a new extent, Connect:Enterprise automatically switches to the next eligible (currently allocated) VLF within the VLFROTAT range.
VLFROTAT=1|n	Specifies the number of VLF files (from 1 to 8) that are eligible for automatic collection. Specifying the value 1–8 places the VLF files into the rotation scheme. When the last defined VLF fills up to the value specified in VLFPCT, the log file is rotated back to VLF1. Connect:Enterprise places the next log record in VLF1. If no suitable rotate file is found, the log file does not change. All logging in progress is finished in the same log file in which it was started. Only new log records are switched to the new log file.
XAPPCSEC=xxxxxxxx	Specifies the load module name of a user-written APPC Security exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XAPPCWI=xxxxxxxx	Specifies the load module name of a user-written APPC Wake Up Initiate exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XAPPCWT=xxxxxxxx	Specifies the load module name of a user-written APPC Wake Up Terminate exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XENDOFB=xxxxxxxx	Specifies the load module name of a user-written End Of Batch exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XEOBVER=2	Indicates system conforms to the programming standards of Connect:Enterprise version 1.2 or later regarding the use of the End-of-Batch Exit programs STEOBX and STEOBX2.
XINIT=xxxxxxxx	Specifies the load module name of a user-written Initialization exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XINPUT=xxxxxxxx	Specifies the load module name of a user-written Input exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XLOG=xxxxxxxx	Specifies the load module name of a user-written Log exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XOUTPUT=xxxxxxxx	Specifies the load module name of a user-written Output exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.

Parameter	Description
XSECUR1=xxxxxxxx	Specifies the load module name of a user-written Security exit one. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XSECUR2=xxxxxxxx	Specifies the load module name of a user-written Security exit two. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
X_SECURE=xxxxxxxx	Specifies the load module name of a user-written FTP session security exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.
XTERM=xxxxxxxx	Specifies the load module name of a user-written Termination exit. Place the load module in the JOBLIB or STEPLIB for Online Connect:Enterprise.

For more information on customizing user exits and using the sample user exit programs related to the user exit ODF *OPTIONS parameters, for example, STSSEC1 for XSECUR1, see the *Connect:Enterprise for z/OS for z/OS Application Agents and User Exits Guide.*

Chapter 4

Configuring ODF Records for SNA Connections

SNA communications sessions with Connect:Enterprise for z/OS can be host-initiated connections to remote sites (Auto Connect sessions) and remote-initiated connections to the Connect:Enterprise for z/OS host. Remote-initiated SNA connections are unsolicited connections with Connect:Enterprise. No action by Connect:Enterprise causes these connections to occur. For these connections to be successful, the Connect:Enterprise ODF must be configured with parameters to accept the remote-initiated connections, and the remote site must supply connection parameters that are acceptable by the Connect:Enterprise host. See *Defining *REMOTES Records for SNA Connections* on page 68 for more information on these parameters.

SNA Auto Connect sessions are initiated by Connect:Enterprise with a remote SNA site. The connection occurs without action by the SNA remote site. Auto Connect sessions can be initiated by the Connect:Enterprise host when data is available for transmission, manually when a connection is desired, or on a scheduled basis. The Connect:Enterprise host is entirely responsible for initiating the Auto Connect session. ODF parameters specify configuration definitions for the external VTAM network that Connect:Enterprise uses to make SNA connections. See *About Auto Connect Sessions* on page 78 for detailed information on Auto Connect sessions and the records used to configure them.

Parameter definitions for remote-initiated sessions and Auto Connect sessions are very similar. The sections that describe configuring ODF records for SNA connections differentiate records and parameters that apply exclusively to remote-initiated sessions or Auto Connect sessions, where applicable.

Defining *OPTIONS Parameters for SNA Connections

The SNA parameters set in the *OPTIONS record enable SNA communications and set global, default values for both host and remote-initiated SNA connections. These values define the attributes of SNA connections unless they are overridden by equivalent parameters set in other records, from the command line, or from one of the user interfaces.

Before you begin configuring the parameters for SNA connections, review the *OPTIONS record format and the rules for defining *OPTIONS parameters in Chapter 3, *Configuring *OPTIONS Record for System Resources*. Required parameters are listed in bold first in the table; the remaining parameters are listed alphabetically.

Parameter	Description
APPLID=xxxxxxxx | ENTPRS	1–8 character ACB name Connect:Enterprise uses to communicate with LU1 devices. If not supplied, the default value, ENTPRS, is used. The name supplied must match the Application Program name defined to VTAM in the ACBNAME parameter of the APPL statement.
LOGONMSG='xxx...xxx' | NO	Supplies a message up to 60 characters that is sent to a remote site's console display screen after a successful logon to Connect:Enterprise. This message is sent only if the remote site can accept it. If this parameter is omitted, the following default message is used: `Connect:Enterprise LOGON COMPLETE` ◆ NO—Specifies that no message is sent to a remote site after a successful logon to Connect:Enterprise.
MAXRWAIT=HH:MM:SS,C	This parameter specifies the maximum Connect:Enterprise wait and retry cycle allowed for the $$REQUEST WAIT= option. Remote sites can specify the amount of time to wait for transmittable batches if no batches are found. The MAXRWAIT parameter provides for a system-wide limit to the WAIT=time specified by any remote site. If not specified, remote sites can specify any valid WAIT= value. ◆ HH:MM:SS—Specifies the amount of time to wait in hours, minutes, and seconds. The maximum allowable in this form is 23:59:59. ◆ C—Specifies the maximum number of wait and retry cycles permitted for any user. The maximum number is 999.
SCINCOR=YES | NO	If SECURITY=BATCH, this parameter must specify whether the Mailbox IDs are maintained in memory or read from the ODF for each Mailbox ID validation. ◆ YES—The Mailbox IDs are in memory. Each ID requires only 8 bytes of storage. This is the recommended value. ◆ NO—The Mailbox IDs are read from the ODF.

Parameter	Description
SECURITY=BATCH | LOGON	When Connect:Enterprise connects with SNA sites, it can specify two SECURITY commands: SECURITY=BATCH and SECURITY=LOGON. When connecting to BSC sites, it can specify only BATCH. ◆ BATCH—Ensures that all transactions transmitted from remote terminals are processed only if a valid Mailbox ID is supplied by the remote site as part of the transmission. Valid IDs are listed in *Defining *SECURITY Record Parameters for SNA Connections* on page 67. ◆ LOGON—Ensures that all logons from remote sites are checked for a valid LU name and are rejected if the LU name is incorrect. Valid LU names are listed in the *POOLS or *REMOTES records in either LUNAME= or RMTACB= in the ODF.
SNA_DEFAULT_$$DIR_FORMAT = BID24 \| BID64	Specifies how Connect:Enterprise formats the reply to a $$DIR command during an SNA session. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. ◆ BID24—The directory display is generated, using the first 24 characters of the User Batch ID. ◆ BID64—The directory display is generated, using the full 64 characters of the User Batch ID. This value can be overridden on a per command basis, by specifying the FORMAT=BID24 | BID64 parameter in the $$DIR command record or by specifying the $$DIR_FORMAT=BID24 | BID64 parameter in the SNA *REMOTES definition.
VSESSLIM=<u>08</u> | nn	Enables you to limit the number of concurrent sessions initiated by remote sites with Connect:Enterprise. (To ensure efficiency, it is necessary to limit sessions during peak usage hours.) This value does not apply to host-initiated Auto Connect sessions. The maximum value is 99. Zero specifies unlimited sessions.
VTAM=YES	Required to activate the VTAM telecommunications method.

Defining *SECURITY Record Parameters for SNA Connections

The *SECURITY record is not required for remote SNA connections unless you define SECURITY=BATCH in the *OPTIONS record, which enables you to implement batch security for SNA connections without implementing the Connect:Enterprise for z/OS security interface. If you are implementing batch security, you must include the 80-byte *SECURITY record immediately following the *OPTIONS record. The *SECURITY record lists the valid Mailbox IDs for your system and restricts remote users to the Mailbox ID assigned to their site.

If you do not implement batch security for SNA connections, you can go to *Defining *REMOTES Records for SNA Connections* on page 68.

*SECURITY Record Format and Parameters

The *SECURITY record is followed immediately by records containing the valid Mailbox IDs for your system. The Mailbox ID is 1–8 characters with no embedded blanks. The following example is a valid ID specification:

```
*SECURITY
       ID=xxxxxxxx
```

The following example shows a record with multiple IDs separated by commas.

```
*SECURITY
       ID=xxxxxxxx,ID=xxxxxxxx,ID=xxxxxxxx
```

*SECURITY Record Rules

When you configure the *SECURITY record, observe the following rules:

- When SECURITY=BATCH in the *OPTIONS record, the *SECURITY record is required and must follow the *OPTIONS record.
- *SECURITY must begin in column 1. Any other text on the same line is ignored.
- Specify multiple Mailbox IDs in a single *SECURITY record.
- Separate Mailbox IDs by either commas or blanks.
- Do not type Mailbox IDs beyond column 72.

Defining *REMOTES Records for SNA Connections

Each SNA site that can establish a session with the Connect:Enterprise host must be defined to Connect:Enterprise for z/OS in the *REMOTES section of the ODF. The *REMOTES record parameters define site-specific values for host-initiated and remote-initiated SNA sessions that override global SNA settings defined in the *OPTIONS record. A remote site can be a terminal, device, or another VTAM application program that Connect:Enterprise considers to be an LU Type 1 RJE device. It can be a remote device on a switched line, a channel-attached local device, or a VTAM application program running on the same or a different host computer.

All remote sites defined to Connect:Enterprise have a remote name. The remote name for an SNA site is supplied to Connect:Enterprise as the DATA portion of a logon to Connect:Enterprise through VTAM. The remote name then accesses a table of information that is built from the *REMOTES record for the SNA site.

The *REMOTES record enables you to:

- Supply options unique to a remote site

- Provide associated LU names for optional LOGON security and for Auto Connect session activation
- Provide default batch Mailbox IDs (remote name is used)
- Determine which batches to transmit during an Auto Connect session

*REMOTES Record Rules

When you configure the *REMOTES record, observe the following rules:

- *REMOTES must begin in column 1. Any other text on the same line after *REMOTES is ignored.
- The NAME= keyword is required and must be the first keyword of the record; and any other text on the line with the NAME keyword is ignored.
- The TYPE= keyword is required and must follow the NAME= keyword; any other text on the line with the TYPE keyword is ignored.
- Required keywords must precede the optional keywords.
- Keywords can begin in any column and can include multiple values.
- Optional keywords can be in any order.

SNA *REMOTES Record Format

The *REMOTES record of the ODF is followed by one or more remote site definitions, each beginning with the NAME keyword. Each NAME record is followed by the TYPE=LU1RJE parameter, and then by any additional optional parameters.

The following example shows the record format for SNA parameters defined in the *REMOTES record. Default values are underlined.

```
*REMOTES
NAME=xxxxxxxx
  TYPE=LU1RJE
  BCHSEP=OPT3
  BLKSIZE=nnnn
  COMPRESS=YES | NO
  CONSOLE=YES | NO
  DISCINTV=nnnn
  FMH=YES | NO | X25 | NPP | IE
  GEISMSG
  ***80-byte message*************************************************
  LOGMODE=xxxxxxxx
  LUNAME=xxxxxxxx or RMTACB=xxxxxxxx or POOL=
  MEDIA=CN | PR | PU | EX | BX
  PASSWORD_CASE=UPPER | MIXED | BOTH
  POOL=xxxxxxxx
  QSESS=YES | NO
  RMTACB=xxxxxxxx
  SC=YES | NO | SPC
  TRANSPAR=Y | N
  TRUNC=YES | NO
  USERDATA='up to 27 characters'
  $$DIR_FORMAT=BID24 | BID64
```

SNA *REMOTES Record Parameters

The following table lists *REMOTES parameter definitions specific to SNA connections. These parameter definitions represent site-specific values that override any corresponding global values set in the *OPTIONS record for remote-initiated and host-initiated SNA connections. Required parameters are listed in bold first in the table; the remaining optional parameters are listed alphabetically. The descriptions also indicate a parameter that applies exclusively to a host-initiated connection or a remote-initiated connection.

Parameter	Description
NAME=xxxxxxxx	Required. Indicates the start of a new remote site definition. It provides the remote name by which the remote site is known. The remote site must specify the 1–8 character name when a LOGON to Connect:Enterprise is attempted.
TYPE=LU1RJE	Required. Specifies the type of Logical Unit (LU) for the remote site, and the SNA protocols used by Connect:Enterprise to communicate with the device. This parameter must immediately follow NAME=. TYPE=LU1RJE is currently the only value supported. It specifies LU Type 1 RJE devices, such as IBM 3770s.
BCHSEP=OPT3	Specifies the method Connect:Enterprise uses to separate batches sent to remote sites when multiple batches are sent in a single connection. Batches are not separated. If multiple batches are sent in a single connection, they are concatenated and sent in a single batch. However, the individual batches are not flagged as transmitted until the entire transmission is successfully completed. Ensure remote sites can process concatenated data batches if this option is chosen. If this parameter is not specified, each batch that is transmitted is bracketed with a begin FMH and end FMH. This enables the remote site to treat each batch as a separate file. **Note:** When the KEEPADD parameter is used during an offline add process that transmits multiple batches, the remote site may process the $$ADD cards embedded in the batches as data records under the following conditions: (1) batches are transmitted with SNA using spanned RUs and (2) the SNA BCHSEP=OPT3 parameter is set. To avoid this situation, set SNA BCHSEP=NO.
BLKSIZE=nnnn | RUSIZE	Specifies the maximum size of a block of data sent to a remote site (RUSIZE). Connect:Enterprise permits values of 0–4096. The default value is taken from the RUSIZE in the session bind. This parameter limits the size of the send buffer transmitted to a value smaller than the session RUSIZE. If the Auto Connect session uses NOBATCH=NC, and SC=NO and COMPRESS=NO are specified for this remote, the BLKSIZE value specified here determines the availability of transmittable batches. The default BLKSIZE value for the NOBATCH=NC function is 512.

Parameter	Description
COMPRESS=YES | NO	Specifies whether compression is supported outbound from Connect:Enterprise to the remote. Specifying COMPRESS=NO suppresses outbound data compression for this remote. The default is YES. If a conflict exists between this parameter and the actual bind received, no compression is done for that remote during the session. This parameter affects only outbound data compression and does not preclude the remote from sending compressed data to Connect:Enterprise.
CONSOLE=YES | NO	Indicates whether the remote device has a console display screen that displays various information messages and error messages from Connect:Enterprise. Most LU Type 1 RJE devices have this capability, and its use with Connect:Enterprise is strongly recommended. The default is YES. For Connect:Enterprise-to-Connect:Enterprise sessions, RMTACB is specified; CONSOLE=NO is required. For Connect:Enterprise-to-Connect:Enterprise Gateway sessions, CONSOLE=YES is required if the remote initiates any sessions (Remote Connects) with Connect:Enterprise.
DISCINTV=0 | nnnn	Specifies the disconnect interval, from 0-3600 seconds. If no session activity occurs for the number of specified seconds, Connect:Enterprise forces the session to end. This feature is used only if your remote site device does not have a similar feature that controls it. It is more efficient for individual remote sites to monitor the session for activity than to have Connect:Enterprise keep track of a variety of DISCINTV values for many active sessions. The default value is 0, indicating that Connect:Enterprise does not force a disconnect if no session activity occurs. An Auto Connect session can override this parameter in the *CONNECT section. Because Auto Connect sessions are host-initiated rather than remote-initiated, the default value differs since the host controls the progress of an Auto Connect session.
FMH=YES | NO | X25 | NPP | IE	Specifies if LU1 3770 FMH is used. The default is YES. ◆ YES—The standard LU1 3770 FMH support is used. ◆ NO—For special use only. Do not specify this value unless directed by Field Level Support. ◆ X25—The LU1 3767 protocol is used. Use this value only for remote sites attached through IBM (NPSI). ◆ NPP—A proprietary protocol for the INS/NPP 9.6 network (United Kingdom) is used. ◆ IE—A proprietary protocol used with the IBM Expedite Direct network.

Parameter	Description
GEISMSG xxxx...xxxx	A placeholder parameter, which indicates that the next record contains the alternate Ready for Input message text string used to connect to the UK GEIS network. **Note:** FMH=NO is required for this parameter. This is an 80-byte message so beware of using edit sequence numbers in columns 73–80. Example: ... FMH=NO GEISMSG xxxx...xxxx BLKSIZE=256 ...
LOGMODE=xxxxxxxx	Specifies the LOGMODE used for the session, which only has meaning if Connect:Enterprise is acting as the Secondary Logical Unit (SLU).
LUNAME=xxxxxxxx | [,xxxxxxxx,...]	Identifies 1–6 Logical Unit names for the remote device. More than one LUNAME is possible if the remote is a Multiple Logical Unit (MLU) device. Use one or more blanks or commas to separate multiple LUNAMEs. All LUNAMEs must fit on a single record. **Note:** The LUNAME parameter is mutually exclusive with RMTACB and POOL. You must use the RMTACB parameter when attempting an auto connect to a remote host application such as JESRJE. This parameter is required if SECURITY=LOGON is specified in the *OPTIONS section, since the LUNAME is validated for the remote name when a LOGON is attempted. This parameter is also *required* if the Auto Connect function is used (*CONNECT). For host-initiated LOGON during an Auto Connect session, the first defined LUNAME attempts connection with the remote site. The LUNAMEs must match those defined to VTAM and NCP. LUNAMEs for switched nodes are defined to VTAM in definition statements for switched major nodes. LUNAMEs for other nodes are defined in your NCP Gen.

Parameter	Description
MEDIA=CN | PR | PU | EX | BX	Directs outbound batches to a specific output media on the remote device. If omitted, the remote site must specify where to direct batches. Otherwise, Connect:Enterprise directs batches to the same media used in an inbound request for a batch. The default is CN. ◆ CN—Directs output batches to display on the remote console screen. This option causes Connect:Enterprise to use a X‘15’ (new line) control character as a record separator. ◆ PR—Directs output batches to print on the remote printer. This option causes Connect:Enterprise to use a X‘15’ (new line) control character as a record separator. ◆ PU—Directs output batches to the remote card punch. This option causes Connect:Enterprise to use a X‘1E’ (standard IRS) as a record separator. Connect:Enterprise Gateway remote sites should select this option. ◆ EX—Directs output batches to the remote exchange diskette and uses Transmission Exchange format. This option causes Connect:Enterprise to use a X‘1E’ (standard IRS) as a record separator. ◆ BX—Directs output batches to the remote exchange diskette and uses Basic Exchange format. This option causes Connect:Enterprise to use a X‘1E’ (standard IRS) as a record separator.

Parameter	Description
PASSWORD_CASE=UPPER | MIXED | BOTH	Specifies how passwords are presented to the security package at logon authorization, in terms of case sensitivity. ◆ UPPER—The password is uppercased before it is presented to the security package for logon security authorization. Only the upper cased value is validated by the security package. For example, if the user enters a password value of "MyPass", the value is uppercased to "MYPASS". If a new password is provided in the logon request, it is also uppercased. ◆ MIXED—The password is not uppercased before it is presented to the security package for logon security authorization. Only the original password value, as entered by the user, is validated by the security package. ◆ BOTH—Both mixed and uppercase password values are validated by the security package, if necessary. First, the original value (mixed case) is validated by the security package. If the logon check fails using the mixed case password, a second attempt is made to validate the password using the uppercased value. A successful logon check with either mixed or upper cased password is considered valid. The intended purpose of specifying BOTH is to allow a transition period for some duration after your systems programmer / security administrator turns on mixed case password support in the security package. This enables Connect:Enterprise to successfully allow users/remotes to logon. Eventually, after users have changed the password with mixed case turned on in the security package, you should set PASSWORD_CASE=MIXED. **Note:** When BOTH is specified, if the first attempt fails (mixed case), but the second attempt is successful (uppercase), Connect:Enterprise considers the logon successful and continues processing as normal. However, the security package still posts a security violation console message due to the failure of the first logon check attempt in mixed case. Also be aware that if both attempts fail, the "consecutive unsuccessful password attempts" count, maintained within the security package, is incremented by 2, since two individual calls are made to the security package.
POOL=xxxxxxxx	Identifies the Logical Unit pool name defined in the *POOLS section. Logical unit names are selected from the specified pool during Auto Connect processing. This parameter is mutually exclusive with LUNAME and RMTACB.
QSESS=YES | NO	When QSESS=YES is specified, Connect:Enterprise enables VTAM to queue the session of the remote SLU when it is unable to immediately accept the session (disabled). This parameter is useful when defining intelligent switch line attached devices (for example, AS/400) that can come active to VTAM in a disabled state. There is no time-out value in effect for session startup when this parameter is used. The default is NO. Applies exclusively to host-initiated sessions.

Parameter	Description
RMTACB=xxxxxxxx	Required if the Connect:Enterprise is going to act as the Secondary Logical Unit in a session with a remote site. It is the APPL name of the Primary Logical Unit (PLU) for which a REQSESS is issued. Applies exclusively to host-initiated sessions. **Note:** The RMTACB, LUNAME, and POOL parameters are mutually exclusive. You must use the RMTACB parameter when attempting an Auto Connect to a remote host application such as JESRJE.
SC=YES | NO | SPC	Specifies whether the remote site is a Sterling Connect site. Currently, a Sterling Connect site is one running Connect:Enterprise. ◆ YES—If SC=YES is specified, CMP=Y is implied and CONSOLE=NO is specified. The SCAN=YES facility for $$ADD control records is ignored on remote sites that have SC=YES specified. Connect:Enterprise attempts to establish a proprietary FMH exchange with the remote. If the exchange is successful, Connect:Enterprise can use proprietary data compression algorithms which allow large record transmissions. ◆ NO—The remote site is not a Sterling Connect site. ◆ SPC—If SC=SPC is specified, the remote site is identified as running Connect:Enterprise Gateway (or SPC version 1.4 or later). In this case, COMPRESS=YES and SCAN=YES are not implied. For Connect:Enterprise-to-Connect:Enterprise Gateway sessions, CONSOLE=YES is required if the remote site initiates any sessions (remote connections) with Connect:Enterprise.
TRANSPAR=Y | N	Specifies whether Connect:Enterprise sends MEDIA=PU batches in transparent mode. ◆ Y—Sends data transparently to the remote site if any characters are found less than x ‘40’. ◆ N—Sends the batch non-transparent using normal x’1E’ record separators regardless of the data content. Transpar=N should only be selected if the data should always be nontransparent to the remote site.
TRUNC=YES | NO	Specifies whether Connect:Enterprise truncates all trailing blanks from records prior to data transmission. The default is NO. ◆ YES—All blanks are truncated prior to data compression and data transmission. The remote site must be able to process truncated data. ◆ NO—No blanks are truncated. TRUNC=NO is used when transmitting variable length records; otherwise, the truncated blanks are not recovered at the receiving site. Setting the TRUNC= parameter in the remote site specification record of the *CONNECT section overrides this parameter for the remote site for the specific Auto Connect session.

Parameter	Description
USERDATA=‘xx....xx’	Required for Connect:Enterprise-to-Connect:Enterprise sessions and for JES2 and POWER sessions. For Connect:Enterprise, it specifies the REMOTE definition used (same as on the NAME= operand in the *REMOTES section of ODF for the Connect:Enterprise session partner). For JES2 and POWER, it specifies the REMOTE name and password, with a maximum length of 27 characters. For Connect:Enterprise-to-Connect:Enterprise, the maximum is also 27 characters. Applies exclusively to host-initiated sessions.
$$DIR_FORMAT=BID24 \| BID64	Specifies how Connect:Enterprise formats the reply to a $$DIR command during an SNA session. If this parameter is not specified, the value from SNA_DEFAULT_$$DIR_FORMAT=BID24 \| BID64 in the ODF *OPTIONS is used for the remote. ◆ BID24—The directory display is generated, using the leftmost 24 characters of the User Batch ID. ◆ BID64—The directory display is generated using the full 64 characters of the User Batch ID. This value can be overridden on a per command basis, by specifying the FORMAT=BID24 \| BID64 parameter in the $$DIR command record.

SNA *REMOTES Sample

The following sample illustrates *REMOTES records for four SNA sites.

```
*REMOTES  DEFINE NINE REMOTE SITES (4 SNA SITES) WITH A VARIETY OF OPTIONS
**-------------SNA Remote Client
Definitions---------------------------------------
  NAME=RMT001
    TYPE=LU1RJE
    LUNAME=LUDAL001,LUDAL002,LUDAL003,LUDAL004,LUDAL005
    MEDIA=PU
  NAME=RMT002
    TYPE=LU1RJE
    LUNAME=LUNYC001
    MEDIA=PR
  NAME=RMT003
    TYPE=LU1RJE
    RMTACB=MAILPLU
    USERDATA=’RMTINPLU’
    LOGMODE=RJE3770
    DISCINTV=30
    CONSOLE=NO
    MEDIA=PU
    SC=YES
  NAME=RMT004
    TYPE=LU1RJE
    POOL=POOL1
    DISCINTV=60
```

The SNA sites illustrated in the preceding sample *REMOTES record have the following characteristics:

- Remote device 1 is an LU Type 1 RJE device with multiple logical unit capability. All outbound batches to the device are directed to the device card punch. Other remote options can use default values.
- Remote device 2 is an LU Type 1 RJE device. All outbound batches to the device are directed to the device printer. Other remote options can use default values.
- Remote device 3 is an LU Type 1 RJE device. RMTACB indicates that this remote is used as the SLU in a Connect:Enterprise Auto Connect session. SC=YES indicates that this is a Sterling Connect device. CONSOLE=NO is required for RMTACB.
- Remote device 4 is an LU Type 1 RJE device. LU name pooling determines the LUNAME used during an Auto Connect session, as indicated by the POOL=parameter.

Configuring the *POOLS Record

The *POOLS record is optional. It identifies the pools of Logical Unit names used when Connect:Enterprise initiates an Auto Connect session to SNA remote sites. Using a pool increases the chances for a successful connection even if one LUNAME is busy, because more than one LUNAME can be associated with a remote site. There is no one-to-one correlation between remote name and LUNAME.

LU name pooling is useful when more than one LU name represents the same physical location as is the case with Connect:Enterprise Gateway. LU pooling with Connect:Enterprise Gateway (or SPC version 1.4 or later) requires the use of the SC=SPC parameter in the *REMOTES section. Multiple pools can be defined for use with an unlimited number of LUNAMEs defined to each pool. LUNAMEs can be defined in more than one pool.

Connect:Enterprise uses the pool order when an Auto Connect session starts. Connect:Enterprise selects the first LUNAME in the pool. Connect:Enterprise verifies that an LUNAME is not being used before attempting a SIMLOGON. If the SIMLOGON fails, the next LUNAME in the pool is selected, verified as inactive, and used for another SIMLOGON attempt. This process continues until the SIMLOGON is successful or the pool is exhausted.

You use a pool by specifying the POOL parameter instead of the LUNAME or RMTACB parameter in the *REMOTES section of the ODF.

*POOLS Record Rules

When you define the *POOLS record, observe the following rules:

- *POOLS must begin in column 1. Any other text on the same line after *POOLS is ignored.
- The NAME= keyword must follow the *POOLS record and can begin in any column. Any other text on the same line after the NAME keyword is ignored.
- Each POOL record must contain at least one LU record.

*POOLS Record Format

The *POOLS record is followed by one or more pool definitions, each beginning with the NAME keyword. Each NAME record is followed by a series of LU records. The following example illustrates a *POOLS record:

```
 *POOLS
NAME=POOL1
 LU=LUNAME1 LUNAME2 LUNAME3 LUNAME4
NAME=POOL2
 LU=LUNAME4 LUNAME5 LUNAME6
 LU=LUNAME7 LUNAME8
```

About Auto Connect Sessions

The Auto Connect function enables host-initiated communications with remote sites. An Auto Connect session is either fully automated or manually initiated. Both data transmission and data collection can be performed during an Auto Connect session. For SNA connections, an Auto Connect session can access any remote site or VTAM application program which is considered an LU Type 1 RJE device and is defined in your VTAM network. An Auto Connect Manager (ACM) is responsible for the Auto Connect session, and ACM tasks can be replicated to allow for processing multiple concurrent requests.

Fully automated Auto Connect sessions are activated each day when the system clock reaches the time of day specified in an Auto Connect list. If Connect:Enterprise remains up for multiple days, the Auto Connect session is activated every day when the system clock reaches the specified time. You can also define *CALENDAR records to specify dates and days of the week on which to activate or deactivate Auto Connect processing. See *Configuring *CALENDAR Records* on page 203 for details.

A fully automated Auto Connect session is initiated by a date, day, or time specified in:

- *CONNECT record in the ODF
- User-written CICS API program

After it is set up, a fully automated Auto Connect session does not need operator intervention at the host site or the remote site, if the hardware at both sites can operate unattended. The desired Auto Connect date, day, or time values must be defined before Connect:Enterprise is brought online. When the defined date, day, or time is reached, Connect:Enterprise starts a connection with the remote sites listed in the ODF.

You can also initiate an Auto Connect session manually by using the:

- $$CONNECT console command

- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, using the $$CONNECT command, you can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program. the CICS, and the ISPF interface also enable you to override Auto Connect options set in the ODF. See *Activating and Overriding Auto Connect Sessions Manually* on page 82 for more information on initiating Auto Connects manually.

Auto Connect Processing

During an Auto Connect session, Connect:Enterprise can send batches to the remote site, receive batches from the remote site, or both send and receive in any order. At SNA sites, Connect:Enterprise always attempts to send batches first. Normally, all batches available for transmission to the remote site are sent immediately. Connect:Enterprise then tries to receive batches from the remote site. Connect:Enterprise continues to receive batches until the disconnect interval expires, indicating that the remote site is finished sending, or until the remote site ends the session.

If the remote site rejects the Connect:Enterprise attempt to send batches, Connect:Enterprise instead attempts to receive batches from the remote site. After the batches are received and the disconnect interval expires, indicating that the remote site has finished sending, Connect:Enterprise again attempts to send batches. If the remote site again rejects the attempt to send batches, Connect:Enterprise again attempts to receive until the disconnect interval expires. This cycle repeats for three send/receive attempts; after that, the session is terminated. The Auto Connect report shows a transmit failure for each rejected attempt to send to the remote site. This could occur if the outbound batches were directed to an unavailable remote site printer.

Send Processing

The following table describes the ways to identify batches sent during an Auto Connect session:

Method	Description
Standard Auto Connect	This method first sends batches that match the remote name, then sends batches that match the LISTNAME. These batches are then sent to all remote sites in the Auto Connect list before they are marked **T** (transmitted).
BEGINLIST parameter	Indicates the batch to be transmitted first. Specify the 1-8 character mailbox ID of the batch. This method could send a batch containing the BSC free-form SIGNON.
Auto Connect by IDLIST	Sends only those batches that match the mailbox IDs specified in the list.
$$CONNECT with ID specified	Sends all batches that match the mailbox ID in the $$CONNECT command. If BATCHID is also specified, all batches that match both ID and BATCHID are sent.

Batch Status Flags

Because you would not typically want to send batches multiple times for different Auto Connect sessions to the same remote site nor send batches that are no longer needed, Connect:Enterprise checks certain batch status flags before sending any batches. These batch status flags are the same as those displayed in $$DIRECTORY output, the ISPF interface, the CICS interface, or in offline utility LIST reports. The following criteria are used by Connect:Enterprise in determining whether a batch is transmitted during an Auto Connect session:

- The batch must be marked R (can be requested).
- The batch must not be marked T (already transmitted).
- The batch must not be marked D (delete).
- The batch must not be marked I (incomplete).
- The batch must not be transmit locked (added by the offline utilities with TRANSMITONCE=YES and then transmitted one time).

One exception to these rules enables you to send a batch that would not normally be sent for an Auto Connect session. If you want to force the retransmission of a batch marked T or I, you can enter its specific Mailbox ID and batch number in a $$CONNECT command from the console or through either the ISPF or CICS interface. See *Activating and Overriding Auto Connect Sessions Manually* on page 82.

When an Auto Connect session is activated but no batches meet the criteria for transmission, Connect:Enterprise sends the following message to the remote site:

```
*** NOTE *** TRANSMIT FAILED NO BATCHES FOR TRANSMISSION
DURING CONNECT:ENTERPRISE AUTO CONNECT.
```

The remote site still has the opportunity to send batches to Connect:Enterprise. For BSC sites, the remote site still has the opportunity to send batches if the MODE includes a RECV.

The NOBATCH=NC option in the Auto Connect list does not attempt a connection and does not send messages if no batches are available for transmission. The NOBATCH=NC feature is implemented for FTP Auto Connect sessions by the code in the LOGON_SCRIPT. See example member NOBATCH for sample REXX code.

Receive Processing

When Connect:Enterprise is receiving batches during an Auto Connect session, the remote site controls what constitutes a batch by the standard Connect:Enterprise $$ADD record. The Mailbox ID specified on the $$ADD from the remote site does not have to match the remote name. However, if Connect:Enterprise batch security is used, the Mailbox ID must be valid. Data received by Connect:Enterprise without a $$ADD record during an Auto Connect session uses the following default values:

```
ID=Remote Name from Auto Connect list
BATCHID=”AC BATCH WITHOUT $$ADD”
XMIT=N
```

Auto Connect receive processing is designed to receive data batches from remote sites with the host site initiating the connection. For this reason, the standard remote-initiated requests ($$REQUEST, $$DIRECTORY, and $$DELETE) are ignored during an Auto Connect receive.

At SNA sites, a $$LOGOFF command can be sent to the host if the remote site wants to end the session at any time.

Pending Processing

When Connect:Enterprise tries to start an Auto Connect session, it is possible that some remote sites in the Auto Connect list are in use by usual remote-initiated calls to the host site. If this is the case, Connect:Enterprise flags the required remote sites as pending Auto Connect sessions. As the remote sites become available, the Auto Connect list begins processing them. Keep in mind that excessively large remote-initiated processing can delay Auto Connect sessions in some cases.

A single remote site can never be shared by two separate Auto Connect sessions, so a pending state is not entered if a remote site is in use by another Auto Connect list. Any Auto Connect sessions that fail due to this condition display a console error message and are reported as failures in the Auto Connect report.

No pending condition is entered if you attempt to start more than one Auto Connect session for a list name which is already active. An attempted Auto Connect start for a list name that is in use fails and an appropriate error message is displayed unless Auto Connect queuing is in use for that list name.

Queuing and Reactivating an Auto Connect Session

When an Auto Connect session cannot start, Connect:Enterprise queues the Auto Connect list and attempts to start it at a later time when its chance of success is greater. Auto Connect queuing activity is logged and reported with the REPORT utility.

Queuing is controlled by parameters set in the *OPTIONS record and the *CONNECT record. Setting ACQDEFAULT=Y in the *OPTIONS record activates queueing for all Auto Connect lists. You can change this default setting for an Auto Connect list by defining the ACQUEUE= parameter in the *CONNECT record for a specific Auto Connect list.

SNA Auto Connect lists are queued when:

- An SNA Auto Connect list is already running.
- An SNA Auto Connect list cannot establish a session with any remote site and the Auto Connect failure code is one of the following:
  - 41 - The host LOGON attempt is rejected by VTAM.
  - 43 - The remote site is not available.
  - 50 - The remote site is already in session for a previous Auto Connect session.

The following table describes the conditions under which a queued SNA Auto Connect list is requeued or reactivated.

If Auto Connect list is queued because	Then it is requeued or reactivated when

ACQUEUE=Y
ACQDEFAULT=Y
	Overriding values are specified (for example, a $$CONNECT command ($$CON) is issued for the same list name but specifying a different mailbox ID)

An Auto Connect session is already running	Reactivated when the previous Auto Connect session ends
No SNA sessions can be established	Requeued and retried in 10-minute intervals until the Auto Connect list can establish a session with at least one remote site, or until 24 hours have expired. After the 24-hour limit is reached, the Auto Connect list is automatically deleted from the queue.

Activating and Overriding Auto Connect Sessions Manually

You can initiate an Auto Connect session manually by using the:

- $$CONNECT console command
- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, $$CONNECT can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program or the CICS or ISPF interface also enables you to override Auto Connect options set in the ODF. The manually activated command is useful if the data is not ready when the fully automated Auto Connect session starts. The type of Auto Connect session initiated depends on the operands used with the **$$CONNECT** command. The following example initiates a full Auto Connect session:

```
$$CONNECT L=LISTNAME
```

Auto Connect sessions can be manually activated at any time by entering the **$$CONNECT** command at an operator console, through the CICS interface or ISPF interface, or through a user-written CICS API program. You can type the following command on the system console or use the ISPF interface or CICS interface to initiate a partial Auto Connect session for a single mailbox ID:

```
$$CONNECT L=xxxxxxxx ID=xxxxxxxx
```

Fully automated Auto Connect sessions process all remote sites in the *CONNECT list and send all batches with a mailbox ID matching the remote name and list name, or the ID in the IDLIST parameter. However, you can use the $$CONNECT command to send a batch with a different mailbox ID to sites on an Auto Connect list, as illustrated in *Sending a Batch with a Different Mailbox ID to SNA Sites* on page 83.

Sending a Batch with a Different Mailbox ID to SNA Sites

Use the ADD utility to add a batch to the VSAM batch files at the host site with a special mailbox ID, such as ID=ALERT, and specify the batch is available for multiple transmissions to several remote sites (MULTXMIT=YES). The following example sends this batch to all remote sites on the east and west coast:

```
$$CONNECT L=ECOAST ID=ALERT
$$CONNECT L=WCOAST ID=ALERT
```

See the *Console Commands* chapter in the *Connect:Enterprise for z/OS for z/OS User's Guide* for a description of the $$CONNECT console command. See the *Connect:Enterprise for z/OS for z/OS CICS User's Guide* to use the Auto Connect feature with CICS. See the *Connect:Enterprise for z/OS for z/OS ISPF User's Guide* to use the Auto Connect feature with ISPF.

Logging and Reporting Auto Connect Activity

Connect:Enterprise maintains a record of all batches sent and received during each Auto Connect session. As an Auto Connect session progresses, log records that describe the activity during the Auto Connect session are created in the VSAM log file. Auto Connect activity is reported by report utilities. The REPORT function in the offline utilities creates reports of activity during an Auto Connect session. The report utilities can run while Connect:Enterprise is online, and you can specify the type of data that is displayed on the report. The following table describes the contents and types of Auto Connect reports that are created.

Record	Description
Summary	Created for each Auto Connect session. The record contains information on the Auto Connect session, such as time and date started, time and date completed, number of successful batches transmitted and collected, and number of failed batches attempted. If an entire Auto Connect session fails, a failure reason code is recorded. If an entire Auto Connect session does not fail but one or more of the detail records have a failure code, the failure code from the first detail record is recorded in the summary record.
Detail	Created for each individual batch sent or received during the Auto Connect session. The record contains information for a single batch, such as time and date started, time and date completed, block count, remote name, Mailbox ID, user batch ID, and batch number. If any errors occur during the batch processing, a failure reason code is recorded.
Queued	Created if an Auto Connect session is queued. The record contains information on the Auto Connect session, reason for queuing, and the time and date it was queued and reactivated. In addition, summary and detail records are written if no SNA session could be established. Use these records to determine if you must take corrective action before the automatic reactivation of the Auto Connect session

Auto Connect Console Messages

A console message is displayed whenever an Auto Connect session is initiated. See the *Connect:Enterprise for z/OS for z/OS Messages and Codes Guide* for descriptions of Auto Connect messages.

If the Auto Connect session cannot start, a console message is issued. This message indicates if the Auto Connect session has been queued or has failed.

For SNA manual dial only, the console operator is prompted by VTAM to dial at the appropriate time. A console message is issued when the Auto Connect session actually gets under way.

When an Auto Connect session ends and all remote sites in the list have been accessed, a series of summary messages are written to the system console indicating the number of successful and failed transmissions and collections.

The REPORT function in the offline utilities enables you to analyze the Auto Connect session and determine what action is needed.

Configuring the *CONNECT Record for SNA Auto Connect Lists

The *CONNECT record implements the Connect:Enterprise Auto Connect function. The *CONNECT record consists of the following components: list name, list type, Auto Connect parameters, and remote site specification records. The *CONNECT parameters specify the name and type of the list, and processing options for the Auto Connect session, such as time to initiate the session, number of concurrent sessions, and queueing. The remote site specification records used with the *CONNECT record specify the remote site, or sites, to contact and enable you to override certain site-specific parameters set in the *REMOTES record for the Auto Connect session.

To use the Auto Connect function, specify a single *CONNECT record followed by one or more Auto Connect lists. Each Auto Connect list is referred to by its LISTNAME. You can create an unlimited number of Auto Connect lists, and a single remote site can be included on multiple Auto Connect lists. The following example illustrates the structure of the *CONNECT record.

```
*CONNECT
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
    Auto Connect parameters
      Remote Site specification record
      Remote Site specification record
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
    Auto Connect parameters
      Remote Site specification record
      Remote Site specification record
```

Because Connect:Enterprise accesses the ODF every time the system is brought online, you can modify ODF values before you execute Connect:Enterprise. After Connect:Enterprise is online, you can activate an Auto Connect session by LISTNAME using the $$CONNECT console command, the ISPF interface, or the CICS interface at any time to temporarily override the ODF parameter values.

*CONNECT Record Format for SNA Auto Connect Lists

Before you configure an SNA Auto Connect list, review the rules in **CONNECT Record Rules* on page 85. The following example illustrates the *CONNECT SNA Auto Connect parameters.

```
LISTNAME=XXXXXXXX
TYPE=LU1RJE
ACQUEUE=Y | N
ACSESS#=01|nn
CALENDAR=xxxxxxxx
DELAY=01 |nnnn
DISCINTV=15 | nnnn
MAXRMT=nn
NOBATCH=C | NC
RETRY=0 | nn
TIME=hh:mm
Remote Site Specification Record
```

*CONNECT Record Rules

When you define the *CONNECT record, observe the following rules:

- *CONNECT must begin in column 1; any other text on that line is ignored.
- LISTNAME must be the first keyword; any other text on that line is ignored.
- The TYPE keyword must follow the LISTNAME keyword; any other text on the same line is ignored.
- Keywords can begin in any column and can include multiple values.
- Optional keywords can be in any order.
- To specify multiple values, separate the values by commas or blanks. If the multiple values do not fit in a single control record, repeat the keyword on a new control record.

*CONNECT Record Parameters for SNA Auto Connect Lists

The following table lists the *CONNECT parameter definitions for SNA Auto Connect sessions. Required parameters are listed in bold first in the table; the remaining optional parameters are listed in alphabetical order. Defaults are underlined.

Parameter	Description
LISTNAME=XXXXXXXX	Required. Specifies the 1–8 character name of an Auto Connect list. If this value is defined as lowercase, Connect:Enterprise will force an uppercase value providing a consistent naming convention for duplicate LISTNAME verification.

Parameter	Description
TYPE=LU1RJE	Required. Specifies the type of protocol for the Auto Connect session. This parameter must immediately follow LISTNAME=. SNA connections require TYPE=LU1RJE.
ACQUEUE=Y | N	Indicates whether the Auto Connect session should be queued and started later if the Auto Connect function cannot establish a session with at least one remote site. If you specify N, the Auto Connect function fails if resources are not available at the time it is initiated. If you do not specify a value, the default is determined by the setting of the ACQDEFAULT parameter in the *OPTIONS section of the ODF.
ACSESS#=nn | 01	Specifies the number of concurrent sessions that Connect:Enterprise should initiate for this Auto Connect process. The maximum value allowed is 48. When each remote site is used as a single logical unit by Connect:Enterprise, ACSESS# is set to the total number of lines or connections that are active at one time. When the Auto Connect list contains identically named remote sites that are accessed as multiple logical units, ACSESS# is the total number of active sessions and should be larger than the number of lines or connections. For more information about using MLU Auto Connect lists, refer to the MAXRMT# parameter.
CALENDAR=xxxxxxxx	Points to a calendar used for time-activated Auto Connect sessions.
DELAY=nnnn | 0	Specifies the number of seconds, from 0–9999, for Connect:Enterprise to delay after ending one session and before beginning another session with a remote site in the Auto Connect list. The purpose of the DELAY parameter is to allow enough time for the network to complete its session cleanup, which helps ensure line availability for new Connect:Enterprise connection attempts. Twenty seconds is usually sufficient. If a DELAY value is specified, a nonzero DISCINTV value must also be used for this Auto Connect list.
DISCINTV= 15 | nnnn	Specifies the disconnect interval, from 1–3600 seconds, that Connect:Enterprise waits for a response. If there is no session activity for the number of specified seconds, Connect:Enterprise forces the session to end. This is a safety feature; use it to prevent an Auto Connect session from suspending if a remote site does not respond. The default value is 15 seconds. When 0 is specified, Connect:Enterprise does not force a session to end. Use a value of 0 only if the remote site is guaranteed to end the session.

Parameter	Description
MAXRMT#=nn	Specifies the maximum number of Multiple Logical Unit (MLU) remote sites to activate for this Auto Connect session. It has meaning only for MLU Auto Connect sessions that contain multiple identically named remote definitions in the Auto Connect list. If no value is specified, the ACSESS# parameter value is used as the default. The maximum value is 48. MAXRMT# is set to the total number of lines or connections active at one time. MAXRMT# is less than ACSESS#, since multiple sessions are active for a single physical remote site when the MLU Auto Connect function is used. When MAXRMT# is used for MLU Auto Connect sessions, identical remote names *must be* defined contiguously in the Auto Connect List for any MLU remote sites. This type of Auto Connect session normally uses the IDLIST parameter on the remote site specification record to specify which batches are sent during the Auto Connect session. If MAXRMT# exceeds ACSESS#, it is reduced to the ACSESS# value during Connect:Enterprise startup.
NOBATCH=C | NC	Specifies whether Connect:Enterprise should attempt a connection with a remote site when no batches are ready for transmission. ◆ C—Indicates make a connection regardless. (default) ◆ NC—Indicates no connection when no batches are transmittable. To receive batches from a remote site during an Auto Connect session, specify NOBATCH=C. A connection is made even when no batches are flagged for transmission to the remote. SNA lists using NOBATCH=NC require a BLKSIZE value for each remote site to determine the availability of batches. The actual RUSIZE of the session is not set until the BIND has completed. If SC=NO and COMPRESS=NO are specified on the remote, batch availability is restricted to batches with initial data blocks that fit within the BLKSIZE of the remote. The BLKSIZE value should reflect the expected RUSIZE.
RETRY=0 | nn	Specifies the number of times Connect:Enterprise retries any communication failure. If this parameter is used, it is a numeric value 1–99. This retry value is in addition to any retries already done by BTAM. An excessive number of retries can slow down the Auto Connect process if remote sites do not respond promptly to a call.
TIME=hh:mm | [,hh:mm, ...]	Specifies one or more time-of-day values when Connect:Enterprise automatically activates a full Auto Connect session for the list. Specified as a 4-digit number separated by a colon, and a valid time using a 24-hour clock (for example, 08:00, 14:30). If this parameter is omitted, the only way to activate this list is to use the $$CONNECT console command or the CICS/ISPF interfaces. If the same time is specified for multiple Auto Connect sessions, they are spaced five seconds apart.

Add an SNA Remote Site to an SNA Auto Connect List

Following the Auto Connect session parameters, you must provide one or more remote site specification records. These records list each remote site accessed and additional options for each remote site.

Remote sites are specified for each Auto Connect list by remote name using the remote site specification record. The remote name for a site contacted by an Auto Connect session must match a remote name defined in the *REMOTES section of the ODF. This name is sometimes used as the Mailbox ID of batches transmitted during an Auto Connect session. A single remote site can be included in different Auto Connect lists, with different connection options in each list.

The Auto Connect feature also can access a table built from the *REMOTES records and obtain the LUNAME of the remote site. Connect:Enterprise uses the LUNAME to initiate a session with the correct remote site. The system administrator must specify the correct LUNAME in the ODF; otherwise, the Auto Connect session either fails or establishes a session with the wrong remote site. The LUNAME is specified by either the LUNAME parameter in the *REMOTES section or by the LU parameter in the *POOLS section of the ODF. See *Defining *REMOTES Records for SNA Connections* on page 68 or *Configuring the *POOLS Record* on page 77 for more information.

When a session is established by manual dialup, Connect:Enterprise derives the location of the remote site from VTAM. VTAM instructs the system administrator to place a call if the remote site is on a switched manual dial line.

Note: If you use a manually dialed Auto Connect connection, you may want to limit the number of remote sites in the Auto Connect list to prevent a manual dial remote site from stalling other remote sites in an Auto Connect list.

In addition to the remote name, you can also define the MEDIA parameter in a remote site specification record to send all batches during the Auto Connect session to a specific output media on the remote device, such as a printer. If the MEDIA parameter is not defined in the remote specification record, the value for the MEDIA parameter defined in the *REMOTES record for the remote site is used. You can override the value for the MEDIA parameter for an Auto Connect list with the $$CONNECT host console command, the CICS interface, the ISPF interface, or a user-written API program.

If the IDLIST parameter is specified for a remote site, then only batches with the indicated Mailbox IDs are transmitted to that site. Otherwise, any batches that match the LISTNAME and remote name are transmitted. You can also use a specific Mailbox ID and BATCHID with the $$CONNECT command, the CICS interface, the ISPF interface, or a user-written API program to control what batches are sent to a specific site.

SNA Remote Site Specification Record Format and Rules

The following examples illustrates the format for SNA remote site specification records. Default values for parameters are underlined. When you add an SNA site to an Auto Connect list, you can define values in the remote site specification record that override the same parameters defined in the *REMOTES record to apply specific processing options for the Auto Connect session that differ from those defined in the *REMOTES record. For example, if COMPRESS=YES is set in the *REMOTES record, you can override this value by setting set CMP=NO in the remote site specification record to suppress outbound data compression to this remote site for the Auto Connect session.

```
REMOTE_NAMEBCHSEP=OPT3 CMP=Y|N MEDIA=CN|PR|PU|EX|BX ONEBATCH=YES|NO TRUNC=N|Y
  BEGINLIST=xxxxxxxx
  IDLIST=xxxxxxxx
  ENDLIST=xxxxxxxx
```

When you define SNA remote site specification records, observe the following rules:

- You must include at least one remote site specification record for an Auto Connect list.
- REMOTE_NAME is required and must be the first parameter specified in a remote site specification record.
- The REMOTE_NAME specified for an Auto Connect list must match a remote site name defined in the *REMOTES section of the ODF.
- Specify all optional parameters in any order on the same line as REMOTE_NAME and separate them by one or more spaces.
- The line containing REMOTE_NAME and optional parameters must precede BEGINLIST, IDLIST, and ENDLIST.
- BEGINLIST, IDLIST, and ENDLIST must be specified as the last parameters in an SNA remote site specification record in the following order: BEGINLIST, IDLIST, ENDLIST.

SNA Remote Site Specification Record Parameters

The following table describes SNA remote site specification record parameters. Defaults are underlined. Required parameters are listed in bold first in the table. With the exception of positional parameters, the remaining parameters are listed in alphabetical order. Acceptable abbreviations for parameters are enclosed in parentheses below the parameter in the following table.

Parameter	Description
REMOTE_NAME	Required. A 1–8 character name specifying the REMOTE_NAME for the remote site. This positional parameter must be the first operand on remote site specification records and match or correspond to a REMOTE_NAME defined in the *REMOTES section of the ODF.
BCHSEP=OPT3	Specifies the method Connect:Enterprise uses to separate batches sent to remote sites when multiple batches are sent in a single connection. If multiple batches are sent in a single connection, they are concatenated and sent in a single batch. However, the individual batches are not flagged as transmitted until the entire transmission is successfully completed. Ensure remote sites can process concatenated data batches if this option is chosen. If this parameter is not specified, each batch that is transmitted is bracketed with a begin FMH and end FMH. This enables the remote site to treat each batch as a separate file. **Note:** When the KEEPADD parameter is used during an offline add process that transmits multiple batches, the remote site may process the $$ADD cards embedded in the batches as data records under the following conditions: (1) batches are transmitted with SNA using spanned RUs and (2) the SNA BCHSEP=OPT3 parameter is set. To avoid this situation, set SNA BCHSEP=NO.

Parameter	Description
CMP=Y | N	Specifies whether compression is supported outbound from Connect:Enterprise to the remote site. Specifying CMP=N suppresses outbound data compression for this remote. When the parameter is specified, it overrides the COMPRESS parameter value in the *REMOTES section. If a conflict exists between this parameter and the actual bind received, no compression is done for that remote during the session. This parameter affects only outbound data compression and does not preclude the remote from sending compressed data to Connect:Enterprise.
MEDIA=CN | PR | PU | EX | BX	Directs outbound batches sent during an Auto Connect session to a specific output media on the remote device. This parameter overrides the media selected in the *REMOTES section. ◆ CN—Directs output batches to be displayed on the remote console screen. This option causes Connect:Enterprise to use a X‘15’ (new line) control character as a record separator. ◆ PR—Prints output batches on the remote printer. This option causes Connect:Enterprise to use a X‘15’ (new line) control character as a record separator. ◆ PU—Directs output batches to the remote card punch. This option causes Connect:Enterprise to use a X‘1E’ (standard IRS) as a record separator. Connect:Enterprise Gateway remote sites should select this option. ◆ EX—Directs output batches to the remote exchange diskette and uses Transmission Exchange format. ◆ BX—Directs output batches to the remote exchange diskette and uses Basic Exchange format.
ONEBATCH=YES | <u>NO</u> (OB=Y | <u>N</u>)	Specifies whether only the first batch that meets the transmission criteria be sent. ◆ NO—Specifies that all batches matching transmission criteria are sent. ◆ YES—Specifies that only the first batch matching transmission criteria is sent.
TRUNC=<u>N</u> | Y	Instructs Connect:Enterprise to truncate all trailing blanks from records prior to data transmission. N indicates that no blanks are truncated. Y indicates that all blanks are truncated. This parameter overrides the TRUNC= specified in the *REMOTES section for this remote, for this Auto Connect session. Use TRUNC=N (default from *REMOTES section) when transmitting variable length records; otherwise, the truncated blanks are not recovered at the receiving site.
BEGINLIST=xxxxxxxx	Specifies the first batch sent to a remote. This is in addition to the normal search for batches transmitted. This parameter is valid only when accompanied by IDLIST. If no transmittable batches are found for IDLIST, the BEGINLIST batch is not sent. If specified, place BEGINLIST before the IDLIST parameter in the remote site specification record.

Parameter	Description
IDLIST=xxxxxxxx | [,xxxxxxx,...]	Enables you to specify a list of specific Mailbox IDs transmitted to the remote site during the Auto Connect session. If this parameter is omitted, batches that match the LISTNAME and REMOTE_NAME are transmitted. Specify one or more Mailbox IDs, separated by commas or blanks. Do not enclose the list in parentheses. If all IDs do not fit on a single record, repeat the IDLIST keyword on subsequent records.
ENDLIST=xxxxxxxx	Valid only when accompanied by IDLIST. Transmittable batches identified by ENDLIST are transmitted after IDLIST batches are sent and only if at least one IDLIST batch was actually transmitted. If specified, place ENDLIST after the last iteration of the IDLIST parameter in the remote site specification record. If IDLIST has been specified across multiple input records, ENDLIST *must be* specified after the last iteration.

Sample *CONNECT Records for SNA Auto Connect Lists

The following sample *CONNECT records illustrate various ways to create Auto Connect lists for SNA sites.

```
*CONNECT
  LISTNAME=LIST1
    TYPE=LU1RJE
    TIME=02:00,04:00
    ACSESS#=2
    DISCINTV=15
    NOBATCH=NC
      BOSTON MEDIA=PR
      NEWYORK MEDIA=PR
      ATLANTA MEDIA=PR
      MIAMI MEDIA=PR
  LISTNAME=LIST2
    TYPE=LU1RJE
    RETRY=2
    TIME=08:00,09:00,10:00,11:00,12:00,13:00,14:00
    TIME=15:00,16:00,17:00
    DISCINTV=60
    DELAY=20
      CHICAGO1 IDLIST=PAYROLL1,ACCTPAY1,ACCTREC1
               IDLIST=MEMOS,REPORTS
      CHICAGO2 IDLIST=PAYROLL2,ACCTPAY2,ACCTREC2
      CHICAGO3 IDLIST=PAYROLL3,ACCTPAY3,ACCTREC3
```

In this example, the two SNA Auto Connect lists accomplish the following:

- LISTNAME=LIST1

  LIST1 is for LU Type 1 RJE remote sites in an SNA network. No failure retries are necessary. The Auto Connect feature is activated automatically at 2:00 a.m. and 4:00 a.m. every day. Two concurrent sessions are activated to allow Connect:Enterprise to communicate with two of the sites simultaneously. A disconnect interval ends the sessions if no activity occurs for 15 seconds. If no batches are ready for transmission to the remote site, no connection is attempted. The list contains four remote sites in Boston, New York, Atlanta, and Miami. Connect:Enterprise transmits batches to the sites that match their remote names, and batches that match the list name LIST1. These batches are all directed to the remote site printers. Connect:Enterprise then enables the remote sites to send inbound batches to the host, as long as the remote responds within the 15-second time limit.

- LISTNAME=LIST2

  LIST2 is for LU Type 1 RJE remote sites in an SNA network. Two retries are allowed for any failures. The Auto Connect feature is activated automatically every hour on the hour between 8:00 a.m. and 5:00 p.m. A disconnect interval ends the sessions if no activity occurs for 60 seconds. A 20-second delay is activated after each remote session ends and before attempting a connection with the next remote on the list. The list contains three remote sites in Chicago. A list of specific batches are sent to the three remote sites. Connect:Enterprise then enables the remote sites to send inbound batches to the host, as long as the remote responds within the 60-second time limit.

Individual Remote Processing

You may want to use one Auto Connect list that contains all remote SNA sites in your system. If you create a list that contains all SNA remote sites, you may also want to create a list for each site so that if Auto Connect processing fails for any site, you can retry remote sites individually. To do this, define an Auto Connect list (ALL) containing all SNA remote sites and an Auto Connect list for each remote site. To help keep track of LISTNAME and remote site values, use the remote name as the LISTNAME for an Auto Connect list that contains a single remote site, as shown in the following sample *CONNECT record for SNA sites.

```
*CONNECT
  LISTNAME=ALL
    TYPE=LU1RJE
    TIME=06:00
      MAINST
      MAPLEAVE
      ELMBLVD
  LISTNAME=MAINST
    TYPE=LU1RJE
      MAINST
  LISTNAME=MAPLEAVE
    TYPE=LU1RJE
      MAPLEAVE
  LISTNAME=ELMBLVD
    TYPE=LU1RJE
      ELMBLVD
```

Frequent Host-Initiated Transmissions

The following example shows sample *CONNECT records to use if you frequently send data batches from the host site to remote sites with minimal operator intervention. To implement this type of Auto Connect session, supply numerous TIME values in the Auto Connect list.

```
                            SNA
*CONNECT
  LISTNAME=FREQUENT
    TYPE=LU1RJE
    TIME=08:00 08:30 09:00 09:30 10:00 10:30 11:00 11:30
    TIME=12:00 12:30 13:00 13:30 14:00 14:30 15:00 15:30
    TIME=16:00 16:30 17:00
      BRANCH01
      BRANCH02
      BRANCH03
      BRANCH04
```

Special Host-Initiated Multiple Logical Unit (MLU) Transmissions

The Auto Connect MLU function can be used for remote sites that have MLU capability. This provides multiple simultaneous sessions with the same remote site. The most common use of this feature is directing transmissions to selected output media (print, punch, console, or exchange).

The following sections describe two methods of implementing the MLU feature.

Method 1: Single Remote Site with Multiple LUNAMES

This method is commonly used in large networks where it is not desirable to define each logical unit as a separate remote site. A single remote site is defined in the *REMOTES section of the ODF, with multiple LUNAMEs specified. The *CONNECT section contains multiple identically named remote sites, with a specific MEDIA and IDLIST to direct batches to the proper media.

In the following example, two remote sites are defined (DALLAS and HOUSTON). Each remote site has three LUs defined to allow simultaneous transmissions to three separate media. Only one physical line is to be used for the Auto Connect session, so that a single remote site is serviced at a time (MAXRMT#=1). ACSESS# is set to three to allow three concurrent sessions.

```
*REMOTES
  NAME=DALLAS
    TYPE=LU1RJE
    MEDIA=CN
    DISCINTV=60
    LUNAME=LUDAL01,LUDAL02,LUDAL03
  NAME=HOUSTON
    TYPE=LU1RJE
    MEDIA=CN
    DISCINTV=60
    LUNAME=LUHOU01,LUHOU02,LUHOU03
*CONNECT
  LISTNAME=LIST1
    TYPE=LU1RJE
    DISCINTV=15
    ACSESS#=3
    MAXRMT#=1
      DALLAS MEDIA=EX IDLIST=DALEXCH
      DALLAS MEDIA=PU IDLIST=DALPUNCH
      DALLAS MEDIA=PR IDLIST=DALPRINT
      HOUSTON MEDIA=EX IDLIST=HOUEXCH
      HOUSTON MEDIA=PU IDLIST=HOUPUNCH
      HOUSTON MEDIA=PR IDLIST=HOUPRINT
```

Method 2: Multiple Remote Sites with Different Names

This method is commonly used in small networks, where a large number of *REMOTES definitions is not a concern. Each logical unit is defined in the *REMOTES section of the ODF with a unique name and LUNAME. The *CONNECT section contains each of the defined remote sites and specifies the proper MEDIA.

In this example, three remote sites are defined (DALEXCH, DALPUNCH, and DALPRINT). Each remote site has a single LU defined. ACSESS# is set to three to allow three concurrent sessions. Since the three remote sites are actually a single physical remote site with three MLUs, only a single line or connection is needed to communicate concurrently with the three remote sites. The batches to be sent to these remote sites use the unique remote name, so IDLIST is not needed to specify the batches to be sent.

```
*REMOTES
  NAME=DALEXCH
    TYPE=LU1RJE
    MEDIA=CN
    DISCINTV=60
    LUNAME=LUDAL01
  NAME=DALPUNCH
    TYPE=LU1RJE
    MEDIA=CN
    DISCINTV=60
    LUNAME=LUDAL02
  NAME=DALPRINT
    TYPE=LU1RJE
    MEDIA=CN
    DISCINTV=60
    LUNAME=LUDAL03
*CONNECT
  LISTNAME=LIST1
    TYPE=LU1RJE
    DISCINTV=15
    ACSESS#=3
      DALEXCH MEDIA=EX
      DALPUNCH MEDIA=PU
      DALPRINT MEDIA=PR
```

This method does not identify the defined remote sites as MLU devices to Connect:Enterprise. You must properly define the LUNAMEs to specify the MLU devices. Method 1 with MAXRMT# specified is the preferred method because it gives you more control over the number of lines or connections allocated to an MLU Auto Connect session.

Special Remote Handling

There is no limit to the number of Auto Connect lists, so you can set up remote sites that require special handling. Also, a remote site can be specified in different Auto Connect lists if you want to access the site with different options.

For example, if your remote sites receive and send data during the day and send electronic mail at night to a printer, use the following sample Auto Connect lists for reference:

```
*CONNECT
  LISTNAME=DAYTIME
    TYPE=LU1RJE
      BRANCH01
      BRANCH02
  LISTNAME=NIGHTIME
    TYPE=LU1RJE
      BRANCH01 MEDIA=PR IDLIST=MEMOBR1,MEMOS,ALERT
      BRANCH02 MEDIA=PR IDLIST=MEMOBR2,MEMOS,ALERT
```

Sample SNA Options Definition Files

This section provides samples of ODFs for SNA connections.

Simple SNA Connection

The following example shows a simple connection using SNA. A password is defined to allow remote sites to have access to a full directory list, but all other *OPTIONS use default values. No Auto Connect function and no system security are used. Two remote sites can establish sessions with Connect:Enterprise.

```
*OPTIONS
  VTAM=YES
  APPLID=ENTPRS
  VPF='ENTPRS.VPF'
  PASSWORD=BANANA
*REMOTES
  NAME=RMT001
    TYPE=LU1RJE
  NAME=RMT002
    TYPE=LU1RJE
```

Complex SNA Connection

The following example illustrates the parameters for a more complex SNA connection.

```
*OPTIONS
  VTAM=YES
  APPLID=ENTPRS
  VPF=’ENTPRS.VPF’
  DEFAULT_MODE=BID64
  PASSWORD=AVOCADO
  SECURITY=LOGON
  SECURITY=BATCH
  SCINCOR=YES
  CONSLOG=YES
  LOGONMSG=’YOU ARE NOW LOGGED ON TO Connect:Enterprise’
  XSECUR1=STSEC1
  APPC=YES
  APPCAPPL=APPCMBOX
  CICSAPPL=CICSMBOX
  CICSMODE=LU62
  CICSTR1=CM62
*SECURITY
  ID=RMT001,ID=RMT002,ID=RMT003,ID=RMT004,ID=DALLAS
  ID=ATLANTA,ID=MIAMI,ID=MEMO1,ID=MEMO2
  ID=MEMO3
*POOLS
  NAME=SPCPOOL1
    LU=SPCLU1,SPCLU2,SPCLU3,SPCLU4
*REMOTES
  NAME=RMT001
    TYPE=LU1RJE
    LUNAME=LUDAL001,LUDAL002,LUDAL003
    MEDIA=PU
    CONSOLE=YES
    DISCINTV=20
  NAME=RMT002
    TYPE=LU1RJE
    LUNAME=LUATL001,LUATL002,LUATL003
    MEDIA=PU
    CONSOLE=YES
    DISCINTV=20
  NAME=RMT003
    TYPE=LU1RJE
    LUNAME=LUMIA001,LUMIA002,LUMIA003
    MEDIA=PU
    CONSOLE=YES
    DISCINTV=20
  NAME=RMT004
    TYPE=LU1RJE
    POOL=SPCPOOL1
    SC=SPC
*CONNECT
  LISTNAME=AUTOSEND
    TYPE=LU1RJE
    TIME=02:00
      RMT001 MEDIA=PR IDLIST=MEMO1
      RMT002 MEDIA=PR IDLIST=MEMO2
```

This VTAM system uses a password, 10 Mailbox IDs for batch security, logon security, the use of the console log facility, and Mailbox IDs stored in the core to improve efficiency. The Auto Connect feature can then send to two of the remote sites. A user-supplied Security Exit is invoked before Connect:Enterprise performs its standard security checks.

Three MLU remote sites are defined. They all have console display screens and the capability to end a session if no activity is detected. All batches sent to the remote sites are directed to the device card punch, except during an Auto Connect session, when they are printed. The fourth remote is using Logical Unit name pooling and is connected to a remote site using SPC version 1.4 or higher. An Auto Connect session is used at 2:00 a.m. every day to send any electronic mail memos that have been accumulated for the remote sites. For more information on configuring SNA Auto Connect lists, see *Configuring the *CONNECT Record for SNA Auto Connect Lists* on page 84.

The parameters to define the CICS interface to Connect:Enterprise are also included. These options allow use of the CICS interface and the CICS API transactions.

Chapter 5

Configuring ODF Records for FTP Connections

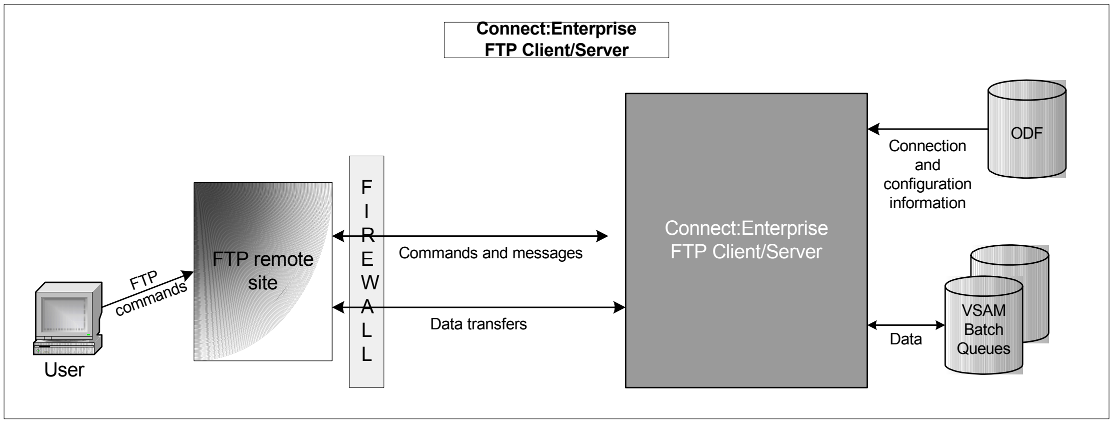
This chapter provides an overview of Connect:Enterprise for z/OS FTP and describes configuring the *OPTIONS record for client and server connections, the*REMOTES parameters for FTP client and server connections, and the *CONNECT record for FTP Auto Connect sessions.

To learn more about the FTP commands available for use from remote sites and how to log on to Connect:Enterprise FTP, see the *Connect:Enterprise for z/OS for z/OS Remote User's Guide*.

Connect:Enterprise FTP

Connect:Enterprise FTP enables Connect:Enterprise to function as an FTP server and client. This enables remote FTP client sites to access, retrieve, and send data to the Connect:Enterprise batch queues through standard FTP commands. This feature also enables Connect:Enterprise to initiate sessions with remote FTP servers (Auto Connect sessions).

The following figure illustrates Connect:Enterprise FTP:


Connect:Enterprise FTP Client/Server
ODF
Connection and configuration information
FTP remote site
F I R E W A L L
Commands and messages
Connect:Enterprise FTP Client/Server
FTP commands
Data transfers
VSAM Batch Queues
Data
User

FTP connection and configuration information is stored in the Connect:Enterprise ODF, just as with SNA and BSC connections. Remote FTP client sites must be defined in the *REMOTES record of the ODF before they can access Connect:Enterprise. Remote FTP server sites must be defined in the *REMOTES record of the ODF before Connect:Enterprise can initiate an Auto Connect session with them. (A special Anonymous remote site definition must be defined to support anonymous FTP; see *Anonymous FTP Remote Site Definitions* on page 133 for more information.) FTP data files are stored and retrieved from the VSAM batch queues.

Security Considerations

You can use the Connect:Enterprise security interface, or a session security exit to enhance security during FTP sessions. Connect:Enterprise also supports both the Transport Layer Security (TLS) protocol and the Secure Sockets Layer (SSL) version 3 protocol to protect data transfers. For more information about using SSL, see *Setting Up Support for SSL Protocol* on page 102, *Configuring *OPTIONS Parameters for FTP Connections* on page 114, and *Configuring *REMOTES Records for FTP Connections* on page 131.

Note: Throughout this chapter, the phrase SSL is used to describe both the SSL and TLS protocols.

Session Security Exit

The session security exit enables users to extend or replace the standard Connect:Enterprise security interface functions. The session security exit is used for FTP sessions only. The session security exit is optional. You must define, code, assemble, link, and test your own session security exit. See the *Connect:Enterprise for z/OS for z/OS Application Agents and User Exits Guide* for more information about the session security exit.

Security Interface Security Check

If you use the Connect:Enterprise security interface, (specified by the MBXSECURE or FTPSECURE ODF *OPTIONS parameter), you must create security rules, and the security system checks if the remote site is authorized to perform the FTP server command. For more information about ways to implement the Connect:Enterprise for z/OS security interface for FTP connections and the ODF *OPTIONS parameters related to security, see Chapter 10, *Implementing the Connect:Enterprise for z/OS Security Interface*. Security interface checks are made only on FTP data transfer commands and the DELE command. If the remote site is authorized to use the command, command processing continues. If the remote site is not authorized to use the command, Connect:Enterprise stops processing the command.

The security check uses the same pseudo data set name to perform FTP security checks as it does for online command checking. (See Chapter 10, *Implementing the Connect:Enterprise for z/OS Security Interface*, for a description of the pseudo data set name.) The following table shows the relationship between the command entered by the Connect:Enterprise FTP client commands, remote $$ commands, and the pseudo data set name. (The FTP Client Command column lists the FTP commands that a remote FTP user enters to invoke the FTP Server Command.)

FTP Client Command	FTP Server Command	$$ Command	Pseudo Data Set Name
delete, mdelete See note.	DELE	$$DEL	<MBXHLQ>.<MBXNAME>.ONLINE.$$DEL.<MBXID>
dir	LIST	$$DIR	<MBXHLQ>.<MBXNAME>.ONLINE.$$DIR.<MBXID>
get, mget See note.	RETR	$$REQ	<MBXHLQ>.<MBXNAME>.ONLINE.$$REQ.<MBXID>
ls	NLST See note.	$$DIR	<MBXHLQ>.<MBXNAME>.ONLINE.$$DIR.<MBXID>
put	STOR	$$ADD	<MBXHLQ>.<MBXNAME>.ONLINE.$$ADD.<MBXID>

Note: Remote users require access to the NLST ($$DIR) command when using the FTP client mdelete or mget commands.

Connect:Enterprise does not perform security checks on any other Connect:Enterprise FTP server commands. However, Connect:Enterprise writes a record of each command to the log file.

Considerations for Configuring the *OPTIONS Record

The *OPTIONS record enables you to define default behavior for firewall navigation, the clear control channel feature, and SSL protocol support. You can use the Connect:Enterprise security interface or a session security exit to further control security during FTP sessions. Also, Connect:Enterprise provides support for Secure Sockets Layer (SSL) version 3 to protect data transfers. The following sections describe configuration options you should consider before you begin configuring the *OPTIONS record for FTP connections.

Assigning an IP Port Number to Connect:Enterprise

You must select the port that Connect:Enterprise monitors when it acts as an FTP server. Note the port number for use when you define the FTP_SERVER_CONTROL_PORT parameter.

You do not need to add parameters to the TCP/IP profile for this definition, unless you choose to do so. If you define the Connect:Enterprise port number in the TCP/IP profile, correctly identify the USERID, procedure name, or job name that the online system will use.

Implementing Firewall Navigation

Certain parameters in Connect:Enterprise for z/OS enable you to implement firewall navigation and apply restrictions to FTP operations in active and passive FTP mode. These parameters enable you to define port ranges, the number of retries if socket acquisition fails, and how long to wait between retries. Although you can specify all available ports on a system in each range, you can control firewall navigation more effectively by assigning a limited number of ports for FTP operations.

To implement firewall navigation, you can specify defaults for up to five ranges of ports and their associated configuration variables in the *OPTIONS section of the ODF. These defaults may be overridden for individual sites defined in the *REMOTES section of the ODF.

Implementing the Clear Control Channel (CCC) Feature

Using the CCC (also known as Clear Command Channel) command provides a way to negotiate the control connection from an encrypted content to a clear text content. After the user ID and password have been transmitted in encrypted format, the remainder of the control transmission is in clear text until the connection ends. All data and objects transferred between the client and server remain encrypted.

Note: Each endpoint of the connection must support the use of this command.

The default policy value for all client or all server connections is set in the *OPTIONS section of the ODF using the SSL_DEFAULT_CLIENT_CCC_POLICY and the SSL_DEFAULT_SERVER_CCC_POLICY parameters. These defaults can be overridden by setting a different value in the SSL_CCC_POLICY parameter for the individual *REMOTES definition. These default CCC policies apply only when SSL=YES.

Setting Up Support for SSL Protocol

As a prerequisite for using the SSL or TLS protocol to secure data during transmission, you must create the SSL key database. You use the IBM utility GSKKYMAN to create the SSL key database. For more information on creating a key database, see the IBM manual *Cryptographic Services System Secure Socket Layer Programming Guide and Reference* (GSKSSLI0).

Your Registration Authority (RA) creates the SSL certificate and the certificate signing request (CSR). The RA forwards the CSR to the selected certificate authority to process the signed certificate.

After you create the key database, you will use the following information to define SSL parameters in the ODF *OPTIONS record:

- Label you assigned to the key
- Path where the key database is stored
- File name of the key database
- Password that was generated for the key database

The following example illustrates configuring SSL parameters for FTP in the *OPTIONS record. The lowercase parameter values are intentional.

```
*OPTIONS
  .
  .
  .
  SSL=YES
  SSL_SERVER_CERT=’server1’
  SSL_CIPHER_SUITE=0A090504
  SSL_KEY_DBASE=’u/user1/user1.kdb’
  SSL_KEY_DBASE_PW=’doyouwanttoknowasecret’
  SSL_TIMEOUT=100
  SSL_DEFAULT_POLICY=OPTIONAL
  SSL_DEFAULT_AUTH_POLICY=OPTIONAL
```

If the Client sends AUTH, followed by PBSZ (Protection Buffer Size) and PROT P (Data Channel Protection Level Private), both the control and data channels are encrypted. If the Client sends AUTH only (no subsequent PBSZ and PROT), Connect:Enterprise for z/OS uses the RFC default of PROT C (data channel in the clear) and does not enforce encryption on the data channel.

Processing $$ADD Commands Embedded in Batches

Batches added to the Connect:Enterprise for z/OS repository often contain embedded $$ADD commands. Connect:Enterprise for z/OS supports scanning for and processing $$ADD commands in batches added to the repository during an FTP session in the following situations:

- When the Connect:Enterprise FTP client issues a RETR command to collect batches from a remote site
- When a remote FTP client connects to the Connect:Enterprise repository and issues a STOR or STOU command to send batches to the repository

For more information on how the remote site should prepare batches for scanning, see the *Connect:Enterprise for z/OS Remote User’s Guide*.

The SCAN= parameter enables Connect:Enterprise for z/OS to process $$ADD commands embedded in batches. When Connect:Enterprise for z/OS processes an embedded $$ADD command, a new batch is created that consists of the data beginning after the $$ADD command and ending at the next recognized $$ADD command or the end of the file. Commands other than $$ADD commands are removed from the file and ignored.

Hierarchy of SCAN Parameters

To process embedded $$ADD commands, scanning must be in effect prior to the receipt of a file. Scanning can be enabled or disabled at the global level, at the remote site level, at the session level, and at the batch level. The following table lists order of precedence, from highest to lowest, in which settings for the SCAN parameter are applied for Connect:Enterprise for z/OS as the FTP client and server:

Hierarchy		Defined in	Parameter
Batch	**Client/ Server**	$$ADD command	SCAN=YES|NO
Session	**Client**	LOCSITE command	LOCSITE SCAN=YES|ALL|NO
	Server	SITE command	SITE SCAN=YES|ALL|NO
Remote Site	**Client**	*REMOTES record for FTP client	SCAN=YES|ALL|NO
	Server	*REMOTES record for FTP server	SCAN=YES|ALL|NO
Global	**Client**	*OPTIONS record	FTP_DEFAULT_CLIENT_SCAN=YES|ALL|NO
	Server	*OPTIONS record	FTP_DEFAULT_SERVER_SCAN=YES|ALL|NO

The values for the SCAN parameters specified in the *OPTIONS record define the default behavior of the Connect:Enterprise for z/OS FTP server and client with regard to processing embedded $$ADD commands. You can override values set at the global level by setting values for individual remote sites in the *REMOTES record. For example, if FTP_DEFAULT_CLIENT_SCAN=NO is set in the *OPTIONS record, but SCAN=YES is set in the *REMOTES FTP server record for a particular remote site, when the Connect:Enterprise for z/OS FTP client initiates an Auto Connect session with this remote site, the SCAN=YES value from the *REMOTES FTP server record overrides the global client setting and enables Connect:Enterprise to scan the received batches for $$ADD commands.

Likewise, when FTP_DEFAULT_SERVER_SCAN=YES is set at the global level, but SCAN=NO is set in the *REMOTES FTP client record for a particular site, when that remote site initiates an FTP connection and issues the STOR or STOU command, Connect:Enterprise for z/OS does not scan the received batches for $$ADD commands because the value set in the *REMOTES record for this site overrides the global value set in the *OPTIONS record.

At the session level, when Connect:Enterprise for z/OS initiates an FTP client connection, the value of the SCAN= parameter set in the script using the LOCSITE command takes precedence over the SCAN parameter values set in the *OPTIONS and *REMOTES records for the duration of the session. For example, if SCAN=NO is set in the *REMOTES FTP server record, but the LOCSITE command sets SCAN=YES, then the Connect:Enterprise FTP client scans the received data for $$ADD commands. Chapter 9, *FTP Auto Connect Scripts*, contains more information about the LOCSITE command and the *Connect:Enterprise for z/OS Remote User's Guide* discusses the SITE command in detail.

The value of the SCAN= parameter set in embedded $$ADD commands overrides all other SCAN parameter values. When Connect:Enterprise for z/OS processes a $$ADD command in the data that has SCAN=NO, scanning stops, and the remainder of the file is considered data.

The other Connect:Enterprise components that affect $$ADD processing are the FTP session security exit and the security interface, both which are discussed in the chapter dealing with online security exits in the *Connect:Enterprise for z/OS Application Agents and User Exits Guide*. Also, refer to *Chapter 10, Implementing the Connect:Enterprise for z/OS Security Interface,* for more information on batch function security. Because embedded $$ADD commands can override the settings for the mailbox ID and batch ID, as well as other flags set by the LOCSITE and SITE commands, both the Connect:Enterprise FTP server and client can call the X_SECURE FTP session security exit as well as the security interface for authorization; however, the client only calls the X_SECURE FTP session exit when scanning is enabled.

Remote Connect:Enterprise for z/OS FTP Client Adds Batches to a Connect:Enterprise for z/OS Server

The following scenario describes the processing of $$ADD commands when a Connect:Enterprise for z/OS remote FTP client issues a STOR or STOU command to transfer batches to a Connect:Enterprise for z/OS server.

Note: In the following examples of $$ADD processing, both the FTP client and server site have Connect:Enterprise for z/OS installed and, therefore, can initiate an Auto Connect session.

When a STOR command is executed and the data being transferred contains embedded $$ADD commands, the dialog between the remote FTP client and the Connect:Enterprise FTP server can be captured as in any trace. The sections of the trace on the client and server sides are different, and examples of each are provided in this discussion. To see the content of individual FTP server and client replies, you can look up a message by its three-digit reply code in *Connect:Enterprise for z/OS for z/OS Messages and Codes Guide.*

The data file used in this scenario contains multiple $$ commands, including the three $$ADD commands highlighted in bold in the following illustration.

```
/*SIGNON    SIGNON1
$$REQUEST   REQUEST1
$$REQ       REQUEST2
$$LOGOFF    LOGOFF1
$$LOG       LOGOFF2
$$DELETE    DELETE1
$$DEL       DELETE2
$$DIRECTORY DIRECTORY1
$$DIR       DIRECTORY2
$$ADD ID=NEWID1 BATCHID='NEW BID 1' TO=Y VBQ#=5 SCAN=YES XMIT=Y $$END
001
$$ADD ID=NEWID2 BATCHID='NEW BID 2' EO=Y VBQ#=4          XMIT=Y $$END
001
002
$$ADD ID=NEWID3 BATCHID='NEW BID 3'      VBQ#=3      MULTXMIT=Y $$END
001
002
003
```

The following REXX script transfers the data file to a Connect:Enterprise for z/OS FTP server using the STOR command. Because the data type is not the default ASCII character set that the

Connect:Enterprise for z/OS FTP server assumes, the data type (TYPE I or Image), data structure (STRU F or File), and transfer mode (Mode C or Compressed) are all specified in the script.

```
/* REXX */
  "LOCCD F34532"
  "CD F34532"
  "SITE SCAN=ALL"
  "LOCSITE BCHSEP=OPT4"
  "MODE C"
  "STRU F"
  "TYPE I"
  "STOR 'F RETR $ M S T E R BFE' 'F RETR $ M S T E R CFI'"
  say "hcrc =" hcrc ", lastrc =" lastrc ", maxrc =" maxrc
  "QUIT"
exit 0
```

The sample STSECFTP exit member was modified using the following code. The exit is called each time a $$ADD command is executed.

```
* ----------------------------------------------------------------34532
         TITLE 'Security for $$ Commands'                         34532
CMD$$$   DS    0H                  Logon Related Command          34532
* ----------------------------------------------------------------34532
*        Place $$ Command Security here                           34532
* ----------------------------------------------------------------34532
         L     R1,E1$XC$PL
         USING X1$DSECT,R1
         L     R2,X1$ID
         L     R3,X1$BCHID
         NI    05(R2),X'CF'
         NI    08(R3),X'CF'
         B     ACT$OK
         DROP  R1
```

In the scenario for this manual Auto Connect session, no overriding values for parameters are specified by *REMOTES parameters. The security parameters are defined in the *OPTIONS record of the ODF, as shown in the following example. The STSECFTP exit is specified as the load module name of the FTP session security exit, and batch security is also defined for the security interface.

```
X_SECURE=STSECFTP
MBXHLQ=EPETE3
MBXNAME=CETC
MBXSECURE=BATCH
```

The following $$CONNECT command starts the Connect:Enterprise for z/OS Auto Connect session from the remote Connect:Enterprise for z/OS FTP client using the REXX script “STOREXEC”.

```
$$CON L=EPETE1,ACSCRIPT=STOREXEC
```

The multipage dialog trace output produced on the Connect:Enterprise for z/OS FTP server for the sample session is divided into sections to illustrate how the processing occurs. The following section shows the dialog between the remote Connect:Enterprise for z/OS FTP client and the Connect:Enterprise for z/OS FTP server that establishes the connection and the actual STOR command.

```
      13:55:13:93              DATE:  2005203
      13:55:13:93  FTP CLIENT INPUT:  CWD F34532
      13:55:13:93  FTP SERVER OUTPUT: 250 CWD was successful. Current working Mailbox is "F34532  ".
      13:55:13:94              DATE:  2005203
      13:55:13:94  FTP CLIENT INPUT:  SITE SCAN=ALL
      13:55:13:94  FTP SERVER OUTPUT: 200 SITE command was accepted.
      13:55:13:97              DATE:  2005203
      13:55:13:97  FTP CLIENT INPUT:  MODE C
      13:55:13:97  FTP SERVER OUTPUT: 200 Data transfer mode is C.
      13:55:13:98              DATE:  2005203
      13:55:13:98  FTP CLIENT INPUT:  STRU F
      13:55:13:98  FTP SERVER OUTPUT: 200 Data structure is F.
      13:55:13:99              DATE:  2005203
      13:55:13:99  FTP CLIENT INPUT:  TYPE I
      13:55:13:99  FTP SERVER OUTPUT: 200 Data representation type is I.
      13:55:14:01              DATE:  2005203
      13:55:14:01  FTP CLIENT INPUT:  PORT 10,20,201,2,9,139
      13:55:14:01  FTP SERVER OUTPUT: 200 PORT request OK (10,20,201,2,9,139).
      13:55:14:01              DATE:  2005203
      13:55:14:01  FTP CLIENT INPUT:  STOR 'F RETR $ M S T E R CFI'.#0000001
      13:55:14:04  PORT RANGE STATUS: STFTPS41         USING LOCAL IPADDR=010.020.201.002,02444/00004
      13:55:14:04  FTP SERVER OUTPUT:
150 Opening data connection. Storing 'F RETR $ M S T E R CFI' as batch number 0000002.
```

As the following trace illustrates, all $$ commands except the $$ADD commands are removed when the Connect:Enterprise FTP server scans the batch. When the retrieval of information begins, the Connect:Enterprise for z/OS FTP server provides the initial setting of the SCAN parameter (SCAN=ALL) and information identifying the batch, including the ID and batch ID taken from the STOR command (Batch created: ID=F34532 BID='F RETR $ M S T E R CFI ' Batch Number=0000002). Next, the FTP server issues a message when a $$ command is recognized to indicate that it is either removing a non-$$ADD command or card (Non-$$ADD command removed: /*SIGNON SIGNON1 and Non-$$ADD command removed: $$REQUEST REQUEST) or executing a $$ADD command ($$ADD command:$$ADD ID=NEWID1 BATCHID='NEW BID 1' TO=Y VBQ#=5 SCAN=YES XMIT=Y $$END).

After all $$ADD parameters are listed, both the X_SECURE FTP exit and security interface are called. Notice that both the mailbox ID and batch ID were changed first by the $$ADD command and then by the security exit. For example, the CWD command changed the current working directory (mailbox) to F34532, the $$ADD command changed it to NEWID1, and the security exit changed it to NEWIDA. Although VBQ 5 was specified for batch 0000002, the first batch collected by the Connect:Enterprise FTP server, VBQ 5 was not available, so batch 0000002 was put on VBQ 1 instead. SCAN=YES is set in this $$ADD command, so scanning continues.

```
      13:55:14:05  0000000 SCAN=ALL is the initial setting
      13:55:14:05  0000000 Batch created: ID=F34532  BID='F RETR $ M S T E R CFI  ' Batch
Number=0000002
      13:55:14:05  0000001 Non-$$ADD command removed: /*SIGNON    SIGNON1
      13:55:14:05  0000002 Non-$$ADD command removed: $$REQUEST   REQUEST1
      13:55:14:05  0000003 Non-$$ADD command removed: $$REQ       REQUEST2
      13:55:14:05  0000004 Non-$$ADD command removed: $$LOGOFF    LOGOFF1
      13:55:14:05  0000005 Non-$$ADD command removed: $$LOG       LOGOFF2
      13:55:14:05  0000006 Non-$$ADD command removed: $$DELETE    DELETE1
      13:55:14:05  0000007 Non-$$ADD command removed: $$DEL       DELETE2
      13:55:14:05  0000008 Non-$$ADD command removed: $$DIRECTORY DIRECTORY1
      13:55:14:05  0000009 Non-$$ADD command removed: $$DIR       DIRECTORY2
      13:55:14:05  0000010 $$ADD command: $$ADD ID=NEWID1 BATCHID='NEW BID 1' TO=Y VBQ#=5 SCAN=YES
XMIT=Y $$END
      13:55:14:05 0000010 $$ADD parameter: ID      = NEWID1
      13:55:14:05 0000010 $$ADD parameter: BATCHID = NEW BID 1
      13:55:14:05 0000010 $$ADD parameter: TO      = Y
      13:55:14:05 0000010 $$ADD parameter: XMIT    = Y
      13:55:14:05 0000010 $$ADD parameter: VBQ#    = 05
      13:55:14:05 0000010 $$ADD parameter: SCAN    = YES
      13:55:14:05 0000010 $$ADD approved  by X_SECURE exit
      13:55:14:05 0000010 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDA
      13:55:14:05 0000010 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID A            '
      13:55:14:06 0000010 $$ADD approved by security interface: EPETE1  ,
EPETE3.CETC.ONLINE.$$ADD.NEWIDA
      13:55:14:06  0000010 $$ADD VBQ05 not available, reverting to VBQ01
      13:55:14:07  0000010 $$ADD Batch re-used: ID=NEWIDA   BID='NEW BID A               ' Batch
Number=0000002
```

Like batch 0000002, batches 0000003 and 0000004 were originally created using the mailbox ID and batch ID in effect at the time the STOR command was issued, but their mailbox IDs and batch IDs were also changed by the security exit before being approved. The message for each $$ command is preceded by a seven-digit number to indicate the instance number of the $$ command being removed or processed. As the following example illustrates, the second $$ADD command executed is actually the 11th command (0000011) recognized by Connect:Enterprise . Each message related to the execution of this command also has 000011 preceding it. These messages occur in real-time and do not have reply codes.

```
     13:55:14:07  0000011 $$ADD command: $$ADD ID=NEWID2 BATCHID='NEW BID 2' EO=Y VBQ#=4
XMIT=Y $$END
     13:55:14:08  0000011 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  ' Batch
Number=0000003
     13:55:14:08  0000011 $$ADD parameter: ID      = NEWID2
     13:55:14:08  0000011 $$ADD parameter: BATCHID = NEW BID 2
     13:55:14:08  0000011 $$ADD parameter: EO      = Y
     13:55:14:08  0000011 $$ADD parameter: XMIT    = Y
     13:55:14:08  0000011 $$ADD parameter: VBQ#    = 04
     13:55:14:09  0000011 $$ADD approved  by X_SECURE exit
     13:55:14:09  0000011 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDB
     13:55:14:09  0000011 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID B           '
     13:55:14:09  0000011 $$ADD approved by security interface: EPETE1  ,
EPETE3.CETC.ONLINE.$$ADD.NEWIDB
     13:55:14:07  0000011 $$ADD Batch re-used: ID=NEWIDB   BID='NEW BID B               ' Batch
Number=0000003
```

```
      13:55:14:11  0000012 $$ADD command: $$ADD ID=NEWID3 BATCHID='NEW BID 3'      VBQ#=3
MULTXMIT=Y $$END
      13:55:14:12  0000012 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  ' Batch
Number=0000004
     13:55:14:12  0000012 $$ADD parameter: ID      = NEWID3
     13:55:14:12  0000012 $$ADD parameter: BATCHID = NEW BID 3
     13:55:14:12  0000012 $$ADD parameter: MULTXMIT= Y
     13:55:14:12  0000012 $$ADD parameter: VBQ#    = 03
     13:55:14:13  0000012 $$ADD approved  by X_SECURE exit
     13:55:14:13 0000012 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDC
     13:55:14:13  0000012 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID C            '
      13:55:14:13  0000012 $$ADD approved  by security interface: EPETE1  ,
EPETE3.CETC.ONLINE.$$ADD.NEWIDC
     13:55:14:142  0000012 $$ADD Batch re-used: ID=NEWIDC   BID='NEW BID C               ' Batch
Number=0000004
```

After all data and replies have been buffered, they are assigned reply codes (226 for successful replies or 426 and 550 for failures). The Connect:Enterprise FTP server has two copies of the same information—one produced real-time without reply codes and the other produced after the STOR command has finished execution with reply codes, as shown in the following illustration. The last 226 message on page 110 indicates that the transfer was complete and also shows the original STOR command, the number of $$ADD commands processed, and the number of bytes of data transferred.

```
      13:55:14:15  FTP SERVER OUTPUT: 226-0000000 SCAN=ALL is the initial setting
      13:55:14:15  FTP SERVER OUTPUT:
226-0000000 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  ' Batch Number=0000002
     13:55:14:15  FTP SERVER OUTPUT: 226-0000001 Non-$$ADD command removed: /*SIGNON    SIGNON1
     13:55:14:15  FTP SERVER OUTPUT: 226-0000002 Non-$$ADD command removed: $$REQUEST   REQUEST1
     13:55:14:15  FTP SERVER OUTPUT: 226-0000003 Non-$$ADD command removed: $$REQ       REQUEST2
     13:55:14:15  FTP SERVER OUTPUT: 226-0000004 Non-$$ADD command removed: $$LOGOFF    LOGOFF1
     13:55:14:15  FTP SERVER OUTPUT: 226-0000005 Non-$$ADD command removed: $$LOG       LOGOFF2
     13:55:14:15  FTP SERVER OUTPUT: 226-0000006 Non-$$ADD command removed: $$DELETE    DELETE1
     13:55:14:16  FTP SERVER OUTPUT: 226-0000007 Non-$$ADD command removed: $$DEL       DELETE2
     13:55:14:16  FTP SERVER OUTPUT: 226-0000008 Non-$$ADD command removed: $$DIRECTORY DIRECTORY1
     13:55:14:16  FTP SERVER OUTPUT: 226-0000009 Non-$$ADD command removed: $$DIR       DIRECTORY2
      13:55:14:16  FTP SERVER OUTPUT:

226-0000010 $$ADD command: $$ADD ID=NEWID1 BATCHID='NEW BID 1' TO=Y VBQ#=5 SCAN=YES XMIT=Y $$END
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: ID      = NEWID1
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: BATCHID = NEW BID 1
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: TO      = Y
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: XMIT    = Y
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: VBQ#    = 05
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD parameter: SCAN    = YES
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD approved  by X_SECURE exit
     13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDA
      13:55:14:16  FTP SERVER OUTPUT:
226-0000010 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID A               '
      13:55:14:16  FTP SERVER OUTPUT:
226-0000010 $$ADD approved  by security interface: EPETE1  , EPETE3.CETC.ONLINE.$$ADD.NEWIDA
      13:55:14:16  FTP SERVER OUTPUT: 226-0000010 $$ADD VBQ05 not available, reverting to VBQ01
      13:55:14:16  FTP SERVER OUTPUT:
226-0000010 $$ADD Batch re-used: ID=NEWIDA   BID='NEW BID A               ' Batch Number=0000002
      13:55:14:16  FTP SERVER OUTPUT:
```

```
226-0000011 $$ADD command: $$ADD ID=NEWID2 BATCHID='NEW BID 2' EO=Y VBQ#=4          XMIT=Y $$END
      13:55:14:16  FTP SERVER OUTPUT:
226-0000011 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  ' Batch Number=0000003
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD parameter: ID      = NEWID2
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD parameter: BATCHID = NEW BID 2
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD parameter: EO      = Y
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD parameter: XMIT    = Y
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD parameter: VBQ#    = 04
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD approved  by X_SECURE exit
      13:55:14:16  FTP SERVER OUTPUT: 226-0000011 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDB
      13:55:14:16  FTP SERVER OUTPUT:
226-0000011 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID B               '
      13:55:14:16  FTP SERVER OUTPUT:
226-0000011 $$ADD approved  by security interface: EPETE1  , EPETE3.CETC.ONLINE.$$ADD.NEWIDB
      13:55:14:16  FTP SERVER OUTPUT:
226-0000011 $$ADD Batch re-used: ID=NEWIDB   BID='NEW BID B               ' Batch Number=0000003
      13:55:14:16  FTP SERVER OUTPUT:

226-0000012 $$ADD command: $$ADD ID=NEWID3 BATCHID='NEW BID 3'      VBQ#=3      MULTXMIT=Y $$END
      13:55:14:16  FTP SERVER OUTPUT:
226-0000012 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  ' Batch Number=0000004
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD parameter: ID      = NEWID3
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD parameter: BATCHID = NEW BID 3
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD parameter: MULTXMIT= Y
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD parameter: VBQ#    = 03
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD approved  by X_SECURE exit
      13:55:14:16  FTP SERVER OUTPUT: 226-0000012 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDC
      13:55:14:16  FTP SERVER OUTPUT:
226-0000012 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID C               '
      13:55:14:16  FTP SERVER OUTPUT:
226-0000012 $$ADD approved  by security interface: EPETE1  , EPETE3.CETC.ONLINE.$$ADD.NEWIDC
      13:55:14:16  FTP SERVER OUTPUT:
226-0000012 $$ADD Batch re-used: ID=NEWIDC   BID='NEW BID C               ' Batch Number=0000004
      13:55:14:16  FTP SERVER OUTPUT:
226 Transfer complete.  'F RETR $ M S T E R CFI',       3 $$ADD cards,          2,028 bytes.
      13:55:14:28              DATE:  2005203
      13:55:14:28  FTP CLIENT INPUT:  QUIT
      13:55:14:28  FTP SERVER OUTPUT: 221 QUIT command received. Goodbye.
      13:55:14:29  CLOSING TRACE
```

All three batches were transferred successfully because scanning was not turned off. The original SCAN parameter, which was set in the ODF, specified ALL, and none of the $$ADD commands specified SCAN=NO. Two of them had no SCAN setting at all.

Connect:Enterprise for z/OS FTP Client Retrieves a Batch from a Remote FTP Server

In this scenario, the Connect:Enterprise FTP client retrieves (RETR) data from a remote FTP server, so all scanning and storing of data occur on the client side. The file containing the data is the same one used in the first example; that is, the same non-$$ADD commands are included with the three $$ADD commands that add batches to the Connect:Enterprise client data repository. The following REXX script is essentially the same as the script in the first example except that a LOCSITE command is issued to specify 3 as the number of the VBQ file on which batches collected during the session are to be stored.

```
/* REXX */
  "LOCCD F34532"
  "CD F34532"
  "LOCSITE VBQ#=03"
  "LOCSITE SCAN=ALL"
  "SITE BCHSEP=OPT4"
  "MODE C"
  "STRU F"
  "TYPE I"
  "RETR 'F RETR $ M S T E R BFE' 'F RETR $ M S T E R CFI'"
  say "hcrc =" hcrc ", lastrc =" lastrc ", maxrc =" maxrc
  "QUIT"
exit 0
```

The following $$CONNECT command is issued to execute the REXX script and start the Auto Connect session.

```
$$CON L=EPETE1,ACSCRIPT=RETREXEC
```

In this scenario, the remote FTP server dialog trace contains no information about $$ADD commands because all of the processing occurs on the FTP client site. The first section of the dialog trace produced on the Connect:Enterprise FTP client consists of FTP server replies showing that the commands from the client were accepted and that the connection was opened, as shown in the following illustration.

```
      10:31:08:62  COMMAND FROM SCRIPT: LOCCD F34532
      10:31:08:62   050 Local Working MAILBOX_ID is F34532
      10:31:08:63  COMMAND FROM SCRIPT: CD F34532
      10:31:08:63    FTP CLIENT OUTPUT: CWD F34532
     10:31:08:63              DATE:  2005206
250 CWD was successful. Current working Mailbox is "F34532  ".
     10:31:08:64  COMMAND FROM SCRIPT: LOCSITE VBQ#=03
     10:31:08:64   000 Locsite command was accepted.
     10:31:08:65  COMMAND FROM SCRIPT: LOCSITE SCAN=ALL
     10:31:08:65   000 Locsite command was accepted.
     10:31:08:66  COMMAND FROM SCRIPT: SITE BCHSEP=OPT4
     10:31:08:66    FTP CLIENT OUTPUT: SITE BCHSEP=OPT4
     10:31:08:66              DATE:  2005206
200 SITE command was accepted.
     10:31:08:67  COMMAND FROM SCRIPT: MODE C
     10:31:08:67    FTP CLIENT OUTPUT: MODE C
     10:31:08:67              DATE:  2005206
200 Data transfer mode is C.
     10:31:08:68  COMMAND FROM SCRIPT: STRU F
     10:31:08:68    FTP CLIENT OUTPUT: STRU F
     10:31:08:68              DATE:  2005206
200 Data structure is F.
    10:31:08:69  COMMAND FROM SCRIPT: TYPE I
    10:31:08:69   FTP CLIENT OUTPUT: TYPE I
    10:31:08:69           DATE:  2005206
200 Data representation type is I.
    10:31:08:70  COMMAND FROM SCRIPT: RETR 'F RETR $ M S T E R BFE' 'F RETR $ M S T E R CFI'
    10:31:08:70  PORT RANGE STATUS: DATA PORT LISTENING
    10:31:08:70  PORT RANGE STATUS:          ACCEPT DATA CONNECT REQUEST ACTIVE,
IPADDR=010.020.201.002,01101/00002.
    10:31:08:70   FTP CLIENT OUTPUT: PORT 10,20,201,2,4,77
    10:31:08:70           DATE:  2005206
200 PORT request OK (10,20,201,2,4,77).
    10:31:08:70   FTP CLIENT OUTPUT: RETR 'F RETR $ M S T E R BFE'
    10:31:08:74           DATE:  2005206
150 Opening data connection.          1 batch,            1,976 bytes selected.
```

The next section of the Connect:Enterprise for z/OS FTP client trace shows the 000 reply codes that reflect the processing taking place. Notice that batch 0000005 should have been put on VBQ 5 as specified in the $$ADD command, but because VBQ 5 was not available, the batch was collected and put on VBQ 3, as specified in the LOCSITE command. Likewise, the $$ADD command and the X_SECURE exit change the mailbox ID and batch ID for the batches; the changes by the X_SECURE exit have priority.

```
      10:31:08:74  000 0000000 SCAN=ALL is the initial setting
      10:31:08:75 000 0000000 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI   ' Batch 
Number=0000005
      10:31:08:75  000 0000001 Non-$$ADD command removed: /*SIGNON   SIGNON1
      10:31:08:75  000 0000002 Non-$$ADD command removed: $$REQUEST  REQUEST1
      10:31:08:75  000 0000003 Non-$$ADD command removed: $$REQ      REQUEST2
      10:31:08:75  000 0000004 Non-$$ADD command removed: $$LOGOFF   LOGOFF1
      10:31:08:75  000 0000005 Non-$$ADD command removed: $$LOG      LOGOFF2
      10:31:08:75  000 0000006 Non-$$ADD command removed: $$DELETE   DELETE1
      10:31:08:75  000 0000007 Non-$$ADD command removed: $$DEL      DELETE2
      10:31:08:75 000 0000008 Non-$$ADD command removed: $$DIRECTORY DIRECTORY1
      10:31:08:75  000 0000009 Non-$$ADD command removed: $$DIR      DIRECTORY2
      10:31:08:75 000 0000010 $$ADD command:$$ADD ID=NEWID1 BATCHID='NEW BID 1' TO=Y VBQ#=5 SCAN=YES 
XMIT=Y $$END
      10:31:08:75  000 0000010 $$ADD parameter: ID      = NEWID1
      10:31:08:75  000 0000010 $$ADD parameter: BATCHID = NEW BID 1
      10:31:08:75  000 0000010 $$ADD parameter: TO      = Y
      10:31:08:75  000 0000010 $$ADD parameter: XMIT    = Y
      10:31:08:75  000 0000010 $$ADD parameter: VBQ#    = 05
      10:31:08:75  000 0000010 $$ADD parameter: SCAN    = YES
      10:31:08:75  000 0000010 $$ADD approved  by X_SECURE exit
      10:31:08:75  000 0000010 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDA
      10:31:08:75 000 0000010 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID A '
      10:31:08:76 000 0000010 $$ADD approved  by security interface: EPETE1   , 
EPETE3.CETC.ONLINE.$$ADD.NEWIDA
      10:31:08:77 000 0000010 $$ADD VBQ05 not available, reverting to VBQ03
      10:31:08:77 000 0000010 $$ADD Batch re-used: ID=NEWIDA  BID='NEW BID A ' Batch Number=0000005

      10:31:08:82 000 0000011 $$ADD command: $$ADD ID=NEWID2 BATCHID='NEW BID 2' EO=Y VBQ#=4 
XMIT=Y $$END
      10:31:08:84 000 0000011 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI   ' Batch 
Number=0000006
      10:31:08:84  000 0000011 $$ADD parameter: ID      = NEWID2
      10:31:08:84  000 0000011 $$ADD parameter: BATCHID = NEW BID 2
      10:31:08:84  000 0000011 $$ADD parameter: EO      = Y
      10:31:08:84  000 0000011 $$ADD parameter: XMIT    = Y
      10:31:08:84  000 0000011 $$ADD parameter: VBQ#    = 04
      10:31:08:84  000 0000011 $$ADD approved  by X_SECURE exit
      10:31:08:84 000 0000011 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDB
      10:31:08:84 000 0000011 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID B '
      10:31:08:84 000 0000011 $$ADD approved  by security interface: EPETE1   , 
EPETE3.CETC.ONLINE.$$ADD.NEWIDB
      10:31:08:85 000 0000011 $$ADD Batch re-used: ID=NEWIDB BID='NEW BID B ' Batch Number=0000006

      10:31:08:88  000 0000012 $$ADD command: $$ADD ID=NEWID3 BATCHID='NEW BID 3'      VBQ#=3 
MULTXMIT=Y $$END
      10:31:08:89 000 0000012 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI   ' Batch 
Number=0000007
      10:31:08:89  000 0000012 $$ADD parameter: ID      = NEWID3
      10:31:08:89  000 0000012 $$ADD parameter: BATCHID = NEW BID 3
      10:31:08:89  000 0000012 $$ADD parameter: MULTXMIT= Y
      10:31:08:89  000 0000012 $$ADD parameter: VBQ#    = 03
      10:31:08:90  000 0000012 $$ADD approved  by X_SECURE exit
      10:31:08:90  000 0000012 $$ADD modified  by X_SECURE exit:  ID changed to NEWIDC
      10:31:08:90 000 0000012 $$ADD modified  by X_SECURE exit: BID changed to 'NEW BID C '
      10:31:08:90 000 0000012 $$ADD approved  by security interface: EPETE1 
EPETE3.CETC.ONLINE.$$ADD.NEWIDC
      10:31:08:91  000 0000012 $$ADD Batch re-used: ID=NEWIDC  BID='NEW BID C ' Batch Number=0000007
226-Transfer complete from F34532  .  Data connection closing.
226        2,028 bytes transferred for        1 batch.
      10:31:10:06  hcrc = 0 , lastrc = 226 , maxrc = 226
```

The last two 226 replies from the remote server show the connection closing after the successful transfer. The SCAN parameter was enabled by the LOSCSITE command and by the SCAN parameter in the first $$ADD command, and neither of the remaining two $$ADD commands disabled scanning.

```
      10:31:08:62  COMMAND FROM SCRIPT: LOCCD F34532
      10:31:08:62   050 Local Working MAILBOX_ID is F34532
      10:31:08:63  COMMAND FROM SCRIPT: CD F34532
      10:31:08:63    FTP CLIENT OUTPUT: CWD F34532
      10:31:08:63              DATE:  2005206
250 CWD was successful. Current working Mailbox is "F34532  ".
      10:31:08:64  COMMAND FROM SCRIPT: LOCSITE VBQ#=03
      10:31:08:64   000 Locsite command was accepted.
      10:31:08:65  COMMAND FROM SCRIPT: LOCSITE SCAN=ALL
      10:31:08:65   000 Locsite command was accepted.
      10:31:08:66  COMMAND FROM SCRIPT: SITE BCHSEP=OPT4
      10:31:08:66    FTP CLIENT OUTPUT: SITE BCHSEP=OPT4
      10:31:08:66              DATE:  2005206
150 Opening data connection.      1 batch,       1,976 bytes selected.
     10:31:08:74 000 0000000 SCAN=ALL is the initial setting
     10:31:08:75  000 0000000 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  '
Batch Number=0000005
     10:31:08:75 000 0000001 Non-$$ADD command removed: /*SIGNON  SIGNON1
     10:31:08:75 000 0000002 Non-$$ADD command removed: $$REQUEST  REQUEST1
     10:31:08:82  000 0000011 $$ADD command: $$ADD ID=NEWID2 BATCHID='NEW BID 2' EO=Y
VBQ#=4        XMIT=Y $$END
     10:31:08:84  000 0000011 Batch created: ID=F34532   BID='F RETR $ M S T E R CFI  '
Batch Number=0000006
      10:31:08:84  000 0000011 $$ADD parameter: ID    = NEWID2
      10:31:08:84  000 0000011 $$ADD parameter: BATCHID = NEW BID 2
      10:31:08:84  000 0000011 $$ADD parameter: EO    = Y
      10:31:08:84  000 0000011 $$ADD parameter: XMIT   = Y
226-Transfer complete from F34532  . Data connection closing.
226      2,028 bytes transferred for      1 batch.
      10:31:10:06  hcrc = 0 , lastrc = 226 , maxrc = 226
```

When scanning is disabled during RETR or STOR/STOU processing by means other than the SITE or LOCSITE command, messages and codes are written to the dialog trace. None of these messages require action, but they are included in dialog traces so that you can see how the $$ADD commands have been processed. To view the explanation of these messages see Chapter 16, *Diagnostics*.

Recordizing Batches

When scanning is in effect, Connect:Enterprise for z/OS scans files for $$commands as well as /*cards, none of which are included in the record count or data. This includes both the non-$$ADD commands as well as the $$ADD commands. For FTP files with detectable records, the record count is the number of records Connect:Enterprise for z/OS detects using certain guidelines, not the number of VBQ records. (For more information on the record delimiters used by Connect:Enterprise for z/OS, see *Connect:Enterprise for z/OS Remote User's Guide* on how the remote client site should prepare batches to be scanned by Connect:Enterprise for z/OS.)

After Connect:Enterprise separates a batch into records using record separator strings, it removes those strings from the data and sets the Recordized Batch indicator to Yes and the File Structure indicator to No in the VCF record for the batch. To see what these indicators are set to for a certain batch, you can either check the settings for the File Structure and Recordized Batch indicators on the Batch Detail Information screen (Part 2) in the ISPF or CICS interface or the Detail Report for the LIST utility, which is described in the *Connect:Enterprise for z/OS User's Guide*. See either the

Connect:Enterprise for z/OS ISPF User's Guide or *Connect:Enterprise for z/OS CICS User's Guide,* depending on the interface you are using.

For information on setting the KIRN (Keep Input Recsep NL) parameter to preserve record separator strings in incoming batches, and setting the RIFS (Recordize Input File Structure) parameter to maintain the file structure as a non-record-oriented data structure, see either *Configuring *OPTIONS Parameters for FTP Connections* on page 114 or *Configuring *REMOTES Records for FTP Connections* on page 131, depending on the level at which you want to implement the functionality of each parameter. To use the KIRN (Keep Input Recsep NL) parameter, the RIFS (Recordize Input File Structure) parameter must be enabled. In addition, you can use the SITE and LOCSITE commands in FTP Auto Connect scripts to change the KIRN and RIFS parameter settings for both the client and server. (See Chapter 9, *FTP Auto Connect Scripts*, for more information.)

Diagnostics

You can run traces on FTP sessions with the $$TRACE console command, the TRACE *OPTIONS parameter, or through CICS and ISPF panels. Also, you can use the $$DIALOG FTPON console command or the DIALOG_FTP *OPTIONS parameter to trace the conversation between the FTP client and the FTP server, as long as either the client or the server is a Connect:Enterprise system. See Chapter 3, *Configuring *OPTIONS Record for System Resources*, for more information on setting the system-level FTP *OPTIONS parameters to trace FTP sessions. See Chapter 16, *Diagnostics*, for more information about FTP diagnostics. To see the content of individual FTP server and client replies, see *Connect:Enterprise for z/OS Messages and Codes Guide.*

Configuring *OPTIONS Parameters for FTP Connections

The *OPTIONS record contains the parameters that enable and define the default behavior of Connect:Enterprise for z/OS FTP client and server operations. Before you configure the parameters for remote FTP connections, review the *OPTIONS record format and the rules for defining *OPTIONS parameters in Chapter 3, *Configuring *OPTIONS Record for System Resources*.

The following table lists the *OPTIONS record parameters specific to FTP connections. Required parameters are listed in bold first in the table; the remaining parameters are listed alphabetically. *OPTIONS parameters that affect either client or server connections exclusively contain the keyword *client* or *server*; the other parameters set global values for both types of connections.

Parameter	Description
FTP=NO | YES	◆ NO—Specifies FTP is inactive. ◆ YES—Activates FTP. FTP=YES is required to initialize the FTP environment and for all SSL-related parameters.

Parameter	Description
FTP_AC_SCRIPT_DEFAULT= xxxxxxxx|*blank*	Specifies the name of the default Auto Connect session script PDS member. This AC session script is used in the event that a specific AC_SCRIPT is not specified in the *CONNECT remote specification record. This script must be a member in a PDS file allocated to the DD SYSEXEC in the Connect:Enterprise JCL.
FTP_ALLOW_GETBYNBR_DFLAG_ DEFAULT=NO|YES	Specifies if FTP server remotes allow remote clients to retrieve batches by batch number even if the selected batch has been marked delete. Can be overridden by setting the FTP_ALLOW_GETBYNBR_DFLAG parameter in the *REMOTES section of the ODF for the particular FTP client for which you want to allow retrieval of deleted batches. ◆ NO—Do not allow remote clients to retrieve deleted batches. ◆ YES—Do allow remote clients to retrieve deleted batches.
FTP_CLIENT_PASV_DATA_IPADDR= R227 | CPADDR	Specifies whether the Connect:Enterprise FTP client should use the IP address from the PASV 227 reply text or the remote site's control connection IP address when establishing a PASV data connection.
FTP_CONNECT_INTERVAL= 0060 | nnnn	Specifies the maximum number of seconds an FTP remote connection or FTP Auto Connect waits for a successful logon. If the logon does not occur within the specified interval, the connection is dropped.
FTP_DEFAULT_CLIENT_ BCHSEP_NONE_FILENAME_ FORMAT =BID24|BID64	Specifies the format of the filename used by the Connect:Enterprise for z/OS Client STOR or PUT command when BCHSEP=NONE. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. BID24 = Uses the left most 24 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. BID64 = Uses all 64 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. **Note:** If the user batch ID contains one or more embedded blanks, single quotes are used to delimit the beginning and end of the filename.

Parameter	Description
FTP_DEFAULT_CLIENT_ BCHSEP_OPT3_FILENAME_ FORMAT=BID24|BID64	Specifies the format of the filename used by the Connect:Enterprise for z/OS Client STOR or PUT command when BCHSEP=OPT3. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. BID24 = Uses the left most 24 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. BID64 = Uses all 64 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. **Note:** If the user batch ID contains one or more embedded blanks, single quotes are used to delimit the beginning and end of the filename.
FTP_DEFAULT_CLIENT_ CONTROL_PORT_RANGE= nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of ports a Connect:Enterprise FTP client uses to transfer data to a remote server. Ranges contain the lowest to the highest port number available in that range. Separate ranges by commas. May be overridden by setting the FTP_CONTROL_PORT_RANGE parameter for the REMOTE_SERVER definition in the *REMOTES section of the ODF. There is no general default port range. no value—If this parameter is not specified and FTP_CONTROL_PORT_RANGE is not defined in the remote server definition, a port is requested from the TCP/IP stack and is assigned randomly from the pool of available port numbers.
FTP_DEFAULT_CLIENT_DATA_PORT _RANGE=U | nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of ports a Connect:Enterprise FTP client uses to transfer data to a remote server. Ranges contain the lowest to the highest port number available in that range. Separate ranges by commas. Can be overridden by setting the FTP_DATA_PORT_RANGE parameter for the REMOTE_SERVER definition in the *REMOTES section of the ODF. There is no general default port range. ◆ no value—If this parameter is not specified and FTP_DATA_PORT_RANGE is not defined in the remote server definition, a port is requested from the TCP/IP stack and is assigned randomly from the pool of available port numbers. ◆ U—Sets the auto connect client data port number to re-use the client control port number used to log on.

Parameter	Description
FTP_DEFAULT_CLIENT_ LOCDIRFORM= BROWSER | BROWSER64| MBOX_CLIENT | MBOX_CLIENT64| MBOX_ZOS | MBOX_ZOS64| $MIBNSDFXY| UNIX | UNIX64	Specifies the format of a line returned by the Connect:Enterprise FTP client in response to an Auto Connect script LOCDIR command. This parameter defines the default value for each session. An auto connect script can override the value by using a locsite command (i.e. LOCSITE DIRFORM=). The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, MBOX_ZOS is the default for this parameter; if BID64 is specified, MBOX_ZOS64 is used for this parameter. BROWSER = Specifies a format supported by browsers, displaying the first 24 characters of the Batch ID. BROWSER64 = Specifies a format supported by browsers, displaying the full 64 character Batch ID. MBOX_CLIENT = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the first 24 characters of the Batch ID. MBOX_CLIENT64 = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the full 64 character Batch ID. MBOX_ZOS = Specifies the Connect:Enterprise $$DIR format, displaying the first 24 characters of the Batch ID. MBOX_ZOS64 = Specifies the Connect:Enterprise $$DIR format, displaying the full 64 character Batch ID. $MBINSDFXY = Reply format options. You can specify as many options as you want and in any order after the initial $ option. ◆ $ = User-defined format ◆ M = Eight-character character Mailbox ID ◆ B = 24-character Batch ID (BID=xxxx….xxxx) ◆ I = 24-character Batch ID (xxxx….xxxx) ◆ N = Seven-digit batch number (#nnnnnn) ◆ S = Eight-digit file size in number of bytes (CT=nnnnnnnn) ◆ D = Time/date of batch creation (hhmm-yyddd) ◆ F = Batch status flags ◆ X = 64-character Batch ID (BID=xxxx….xxxx) ◆ Y = 64-character Batch ID (xxxx….xxxx) UNIX = Specifies the standard UNIX directory display format, displaying the first 24 characters of the Batch ID. UNIX64 = Specifies the standard UNIX directory display format, displaying the full 64 character Batch ID.

Parameter	Description
FTP_DEFAULT_CLIENT_ REMOTE_FILENAME_LENGTH= SHORT | LONG | LONG64	Specifies the format of the filename created by the Connect:Enterprise for z/OS FTP Client when sending data to the remote FTP server using the STOR or PUT command if the *REMOTES TYPE=FTP_CLIENT REMOTE_FILENAME_ LENGTH parameter is not set. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, LONG is the default for this parameter; if BID64 is specified, LONG64 is used for this parameter. SHORT = Uses the seven-character batch number as the filename format. LONG = Uses the 24 character User Batch ID as the filename format. LONG64 = Uses the 64 character batch User ID as the filename format.
FTP_DEFAULT_CLIENT_SCAN= NO | YES | ALL	Specifies whether the Connect:Enterprise for z/OS FTP client scans RETR received batches for $$ commands and /* cards. NO= Scanning for Connect:Enterprise for z/OS $$ commands is not enabled. Connect:Enterprise for z/OS $$ commands, /*SIGNON, and /*BINASC cards embedded in a received batch are treated as data. YES= Scanning for Connect:Enterprise for z/OS $$ commands is enabled initially, but scanning for a subsequent $$ADD card is not automatic. Each $$ADD card must include the parameter SCAN=YES to continue scanning for $$ commands. Use this value to make FTP command scanning behave like it does in SNA. ALL= Scanning for Connect:Enterprise for z/OS $$ commands is enabled for the entire batch unless the batch contains a $$ADD card with the parameter SCAN=NO. Use this value to make FTP command scanning behave like it does in BSC.
FTP_DEFAULT_DIALOG_ TRACE_LRECL=136|nnnnn	Specifies the logical record length (LRECL) of the FTP DIALOG trace files (136–32756 characters). Each file is allocated using RECFM=VBA (Variable, Blocked, ANSI print control character). The default value is 136.
FTP_DEFAULT_DISCTINV= 900 | 0 | 3600	Specifies the maximum number of seconds an FTP session may be inactive before forcing session termination. Leading zeros are not required. ◆ 0—Indicates there is no disconnect interval and the connection remains open until a normal disconnect via the FTP QUIT command. May be overridden for individual remote definitions by specifying the DISCTINV= parameter in the ODF*REMOTES section.

Parameter	Description
FTP_DEFAULT_KIRN=YES | NO	(KIRN stands for Keep Input Recsep NL) Specifies whether or not Connect:Enterprise for z/OS removes the record separator string when the batch is stored. NO = Connect:Enterprise for z/OS removes the record separator string after recordizing the batch. This is the default. YES = The Record separator strings are kept in the batch. The corresponding FTP_DEFAULT_RIFS parameter must be set to YES. See *Processing $$ADD Commands Embedded in Batches* on page 103 for more information.
FTP_DEFAULT_PORT_RETRIES=nn | 0	Specifies how many times (from 0–99) a socket connection attempt is made for each control port or data port in the defined range or ranges. The default value is zero, or no retries, thus a socket connection attempt is made only once for each defined socket. May be overridden by setting the FTP_PORT_RETRIES parameter in the remote client or remote server definition in the *REMOTES section of the ODF.
FTP_DEFAULT_PORT_RETRY_ WAIT_TIME=nnn | 030	Specifies the number of seconds (from 0–180) the server waits between socket connection attempts. The default value is 30 seconds. May be overridden by setting the FTP_PORT_RETRY_WAIT_TIME parameter in the remote client or remote server definition in the *REMOTES section of the ODF.
FTP_DEFAULT_RECEIVE_ OPTION_RENAME= FIRST24 | LAST24 | FIRST64 | LAST64	Specifies the filename (User Batch ID) used by the Connect:Enterprise for z/OS FTP Server when creating batches sent from the remote FTP client if the *REMOTE TYPE=FTP_CLIENT RECEIVE_OPTION RENAME value is not set. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, FIRST24 is the default for this parameter; if BID64 is specified, FIRST64 is used for this parameter. FIRST24 = Truncates a long file name by using the first 24 characters of the inbound file name as the User Batch ID. LAST24 = Truncates a long file name by using the last 24 characters of the inbound file name as the User Batch ID. FIRST64 = Truncates a long file name by using the first 64 characters of the inbound file name as the User Batch ID. LAST64 = Truncates a long file name by using the last 64 characters of the inbound file name, as the User Batch ID.

Parameter	Description
FTP_DEFAULT_RIFS=YES | NO	(RIFS stands for Recordize Input File Structure) Changes the batch to record structure or retains the batch as file structure. YES = Recordizes the batch after recognizing a record separator string and uses CRLF (x'0D0A) for SFA batches and NL (x'15') for SFE batches. This is the default. NO = Retains file structure of batch and does not recognize record separator strings in SFA or SFE batches. See *Processing $$ADD Commands Embedded in Batches* on page 103 for more information. **Note:** Processing results cannot be predicted or supported when FTP_DEFAULT_RIFS=NO and FTP_DEFAULT_SERVER_SCAN or FTP_DEFAULT_CLIENT_SCAN is set to YES or ALL.
FTP_DEFAULT_SERVER_ BCHSEP_NONE_FILENAME_ FORMAT= BID24 | BID64	Specifies the format of the filename used by the Connect:Enterprise for z/OS Server in response to a NLST command from the remote client when BCHSEP=NONE. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. BID24 = Uses the left most 24 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. BID64 = Uses all 64 characters of the User Batch ID from the first eligible batch in the transmission as the filename format. **Note:** If the user batch ID contains one or more embedded blanks, single quotes are used to delimit the beginning and end of the filename. **Note:** One line item is returned for batches with the same User Batch ID.

Parameter	Description
FTP_DEFAULT_SERVER_ BCHSEP_OPT3_FILENAME_ FORMAT =BID24 | BID64	Specifies the format of the filename used by the Connect:Enterprise for z/OS Server in response to a NLST command from the remote client when BCHSEP=OPT3. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. BID24 = Uses the left most 24 characters of the User Batch ID from the first eligible batch in the transmission as the filename. BID64 = Uses all 64 characters of the User Batch ID from the first eligible batch in the transmission. **Note:** If the user batch ID contains one or more embedded blanks, single quotes are used to delimit the beginning and end of the filename. **Note:** One line item is returned for batches with the same User Batch ID.
FTP_DEFAULT_SERVER_DATA PORT_RANGE=L-1 | nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of ports a Connect:Enterprise FTP server uses to transfer data to a remote client. May be overridden by setting the FTP_DATA_PORT_RANGE parameter in the REMOTE_CLIENT definition in the *REMOTES section of the ODF. There is no general default port range. ◆ no value—If this parameter is not specified and FTP_DATA_PORT_RANGE is not defined in the remote client definition, a port is requested from the TCP/IP stack and is assigned randomly from the pool of available port numbers. ◆ L-1—A special value that sets the data port to the logon listen port number minus one. Used when the server connects back to a known port number on the client. ◆ nnnnn-nnnnn—Specifies a range of port numbers. Ranges contain the lowest to the highest port numbers available in that range. Separate ranges by commas. A single port is designated by setting the same value in both the low and high port number fields.

Parameter	Description
FTP_DEFAULT_SERVER_DIRFORM= BROWSER | BROWSER64 | MBOX_CLIENT | MBOX_CLIENT64 | MBOX_ZOS | MBOX_ZOS64 | $MIBNSDFXY | UNIX | UNIX64	Specifies the format of a line returned by the Connect:Enterprise FTP server to the remote FTP client in response to the LIST command. This parameter defines the default value for each session. A remote FTP client can override the value using a SITE command. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, MBOX_ZOS is the default for this parameter; if BID64 is specified, MBOX_ZOS64 is used for this parameter. BROWSER = Specifies a format supported by browsers, displaying the first 24 characters of the Batch ID. BROWSER64 = Specifies a format supported by browsers, displaying the full 64 character Batch ID. MBOX_CLIENT = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the first 24 characters of the Batch ID. MBOX_CLIENT64 = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the full 64 character Batch ID. MBOX_ZOS = Specifies the Connect:Enterprise $$DIR format, displaying the first 24 characters of the Batch ID. MBOX_ZOS64 = Specifies the Connect:Enterprise $$DIR format, displaying the full 64 character Batch ID. $MBINSDFXY = Reply format options. You can specify as many options as you want and in any order after the initial $ option. ◆ $ = User-defined format ◆ M = Eight-character character Mailbox ID ◆ B = 24-character Batch ID (BID=xxxx….xxxx) ◆ I = 24-character Batch ID (xxxx….xxxx) ◆ N = Seven-digit batch number (#nnnnnn) ◆ S = Eight-digit file size in number of bytes (CT=nnnnnnnn) ◆ D = Time/date of batch creation (hhmm-yyddd) ◆ F = Batch status flags ◆ X = 64-character Batch ID (BID=xxxx….xxxx) ◆ Y = 64-character Batch ID (xxxx….xxxx) UNIX = Specifies the standard UNIX directory display format, displaying the first 24 characters of the Batch ID. UNIX64 = Specifies the standard UNIX directory display format, displaying the full 64 character Batch ID.

Parameter	Description
FTP_DEFAULT_SERVER_REMOTE_FILENAME_LENGTH= SHORT | LONG | LONG64	Specifies the format of the filename created by the Connect:Enterprise for z/OS FTP Server returned in an NLST reply when BCHSEP=OPT4 is used. Specifying this parameter defines the default value to use when the *REMOTES TYPE=FTP_SERVER REMOTE_FILENAME_LENGTH parameter is not set. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, LONG is the default for this parameter; if BID64 is specified, LONG64 is used for this parameter. SHORT = Uses the 7 character batch number as the filename format. LONG = Uses the 24 character User Batch ID as the filename format. LONG64 = Uses the 64 character User batch ID as the filename format.
FTP_DEFAULT_SERVER_SCAN= NO | YES | ALL	Specifies whether the Connect:Enterprise for z/OS FTP server scans STOR/STOU received batches for $$ commands and /* cards. NO = Scanning for Connect:Enterprise for z/OS $$ commands is not enabled. Connect:Enterprise for z/OS $$ commands, /*SIGNON, and /*BINASC cards embedded in a received batch are treated as data. YES = Scanning for Connect:Enterprise for z/OS $$ commands is enabled initially. To keep scanning in effect when a $$ADD card is encountered, each $$ADD card must include the parameter SCAN=YES. Use this value to make FTP command scanning behave like it does in SNA. ALL = Scanning for Connect:Enterprise for z/OS $$ commands is enabled for the entire batch unless the batch contains a $$ADD card with the parameter SCAN=NO. Use this value to make FTP command scanning behave like it does in BSC.
FTP_LOGON_REPLY=0 | nn	Identifies the location of any additional messages lines. A 220-connection response is always issued to an FTP client successfully logging on to a Connect:Enterprise FTP server. Connect:Enterprise issues a sequence of three or more 0220 messages. ◆ 0—No additional responses are specified. ◆ nn—1–99 lines follow this line in the ODF. A series of 220 connection response messages is issued to the client FTP.

Parameter	Description
FTP_LOGON_SCRIPT_DEFAULT= xxxxxxxx	Optional parameter which specifies the name of the default Auto Connect LOGON_SCRIPT PDS member. A remote server definition without a LOGON_SCRIPT specification generates an error if this default value is not set. This default logon script is used when a different script is not specified in the *REMOTES definition. This script must be a member in a PDS file allocated to the DD SYSEXEC in the Connect:Enterprise JCL. **Note:** If the logon script information is contained in the AC_SCRIPT, you must still define this parameter in the ODF; otherwise, you receive an error.
FTP_MAX_CLIENT_THREADS= 10 | nnnn	Specifies the number of session threads available for auto connections after startup. The maximum value is 9999.
FTP_MAX_SERVER_THREADS= 10 | nnnn	Specifies the maximum number of concurrent FTP sessions. The maximum number of concurrent sessions allowed for the Process/Job by UNIX system services is set in the MAXTHREADTASKS parameter specified in SYS1.PARMLIB (PPXPRMxx). The maximum value is 9999, and leading zeroes are not required.
FTP_SERVER_CONTROL_ PORT=['hostname', | 'nnn.nnn.nnn.nnn'] nnnnn	Specifies which TCP/IP control port the FTP Listener/Dispatcher monitors for FTP high, medium, and low priority session requests. ◆ 'hostname'—Optional preceding hostname (IP address). Specifies a *hostname* if predefined in the ODF for the return data connection to bind to. ◆ 'nnn.nnn.nnn.nnn'—Optional preceding IP address. Specifies an IP address if predefined in the ODF for the return data connection to bind to. ◆ nnnnn—The control port number that monitors FTP session requests; default = 5555.
SCRIPT_INTERVAL_TIME=0030 | nnnn	Specifies the interval (in seconds) between host calls in the AC_SCRIPT or LOGON_SCRIPT. Used to prevent looping REXX scripts.
SSL=NO | YES	Required to activate SSL (Secure Sockets Layer) protocol support.

Parameter	Description
SSL_CIPHER_SUITE= 0A09050403020106 | cipher-suite-list	Specifies a character string that contains the list of SSL version 3.0 ciphers in the order of usage preference. Any combination of values is valid, used in any order. More ciphers may be available. Check your level of cryptographic services for available ciphers when making modifications. The values are: ◆ 0—NULLMD5 ◆ 02—NULLSHA ◆ 03—RC4MD5Export ◆ 04—RC4MD5US ◆ 05—RC4SHAUS ◆ 06—RC2MD5Export ◆ 09—DESSHA ◆ 0A—TripleDESHAUS
SSL_DEFAULT_CLIENT_ AUTH_POLICY=OPTIONAL | REQUIRED | DISALLOWED	Specifies whether SSL client authentication is optional, required, or disallowed when not specified on the client remote site. If the client remote name is not yet known, this value is the only source for setting client authentication policy on a session until the client remote name is identified. ◆ OPTIONAL—Specifies that connections between the remote site and Connect:Enterprise *can* be made secure using SSL at the client's discretion. When OPTIONAL is set, the Connect:Enterprise for z/OS FTP server requests the client certificate. If no client certificate is available, the session continues successfully. If a client certificate is available, it is sent to the server for client authentication. If the client authentication fails, the session fails. Therefore, if you do not want the Connect:Enterprise for z/OS FTP server to attempt to perform client authentication, set DISALLOWED. ◆ REQUIRED—Specifies that connections between the remote site and Connect:Enterprise *must* be made secure using SSL. When REQUIRED is set, the Connect:Enterprise for z/OS FTP server requests the client certificate and authenticates it. ◆ DISALLOWED—Specifies that connections between the remote site and Connect:Enterprise *will not* be made secure using SSL. When DISALLOWED is set, the Connect:Enterprise for z/OS FTP server does not request the client certificate.

Parameter	Description
SSL_DEFAULT_CLIENT_CCC_ POLICY=DISALLOWED | OPTIONAL | REQUIRED	Determines the response when Connect:Enterprise acting as a server, receives the CCC command from an FTP client. Sets the default CCC policy for FTP remote clients. May be overridden for specific clients by setting the SSL_CCC_POLICY parameter in a remote client definition. ◆ DISALLOWED—The CCC command is not honored and the control session remains encrypted. This is the default value. ◆ OPTIONAL—The CCC command is honored if the client sends the command. No error results if the client does not send the CCC command. ◆ REQUIRED—The SSL FTP remote client issues the CCC command and Connect:Enterprise must process the CCC command before any data port operation can be attempted
SSL_DEFAULT_POLICY= OPTIONAL | REQUIRED | DISALLOWED	Sets the SSL requirement between the remote client and the server. May be overridden for specific clients and servers by setting the SSL_POLICY parameter in a remote client or server definition. ◆ OPTIONAL—Specifies that connections between the remote site and Connect:Enterprise *can* be made secure using SSL or TLS at the client's discretion. If AUTH SSL or TLS is received, the control channel will be secure. Unless PBSZ:PROT (Protection Buffer Size:Data Channel Protection Level) is received to change PROT (Protection Level) from the default value of C (Clear), data on the data channel will be in the clear. ◆ REQUIRED—Specifies that connections between the remote site and Connect:Enterprise *must* be made secure using SSL or TLS. As client, Connect:Enterprise automatically sends PBSZ 0 and PROT P (Private) to secure both the control and data channels. As server, Connect:Enterprise automatically sets PBSZ 0 and PROT P on both the control and data channels even if these commands are not received from the client . A non-Connect:Enterprise client may still explicitly send the PBSZ and PROT P commands. ◆ DISALLOWED—Specifies that connections between the remote site and Connect:Enterprise will not be made secure using SSL or TLS. Both the control channel and data channel will be in the clear. See *Setting Up Support for SSL Protocol* on page 102 for more information.

Parameter	Description
SSL_DEFAULT_SERVER_CCC_ POLICY=DISALLOWED | OPTIONAL | REQUIRED	Determines the response when Connect:Enterprise acting as a client, initiates a connection to an FTP server. Sets the default CCC policy for FTP remote servers. May be overridden for specific servers by setting the SSL_CCC_POLICY parameter in a remote server definition. ◆ DISALLOWED—The CCC command is not issued by Connect:Enterprise acting as a client. This is the default value. ◆ OPTIONAL—The CCC command is issued and, if accepted, it is honored. If it is rejected, the session remains active and the control connection is encrypted. ◆ REQUIRED—The CCC command is sent. If the remote server rejects the command a QUIT command is issued.
SSL_KEY_DBASE= ’key-data-base-path-name’	Required if SSL=YES. Specifies a character string that identifies the path and file name of the key database file. The key database file is an existing z/OS HFS file. The character string is enclosed in single quotes and can contain embedded blanks, quotes, and any other characters. The outer quotes are used as the string delimiters. The character string can continue over multiple ODF records by specifying multiple consecutive SSL_KEY_DBASE parameters. The strings concatenate to a maximum of 1024 characters. Examples: ◆ SSL_KEY_DBASE=’my/test/path/cert.kdb’ ◆ SSL_KEY_DBASE=’my/long/test/path/name/that/’ SSL_KEY_DBASE=’continues/onto/the/next/ODF/record/cert.kdb’ Resolves to: my/long/test/path/name/that/continues/onto/the/next/ODF/record/cert.kdb **Note:** The SSL_KEY_DBASE=’key-data-base-path-name’, SSL_KEY_DBASE_PW=’key-data-base-password’, and SSL_SERVER_CERT=’certificate-label-string’ parameters must be used together and are mutually exclusive of the SSL_KEYRING_LABEL=’ring-label’ and SSL_KEYRING_NAME=’ring-name’ parameters.

Parameter	Description
SSL_KEY_DBASE_PW= ’key-data-base-password’	Required if SSL=YES. Specifies a character string that identifies the password for the key database file. The character string is enclosed in single quotes and can contain embedded blanks, quotes, and any other characters. The outer quotes are used as the string delimiters. **Note:** The SSL_KEY_DBASE=’key-data-base-path-name’, SSL_KEY_DBASE_PW=’key-data-base-password’, and SSL_SERVER_CERT=’certificate-label-string’ parameters must be used together and are mutually exclusive of the SSL_KEYRING_LABEL=’ring-label’ and SSL_KEYRING_NAME=’ring-name’ parameters. When you print to SYSPRINT, the SSL_KEY_DBASE_PW value is masked. The character string may continue over multiple ODF records by specifying multiple consecutive SSL_KEY_DBASE_PW= parameters. The strings concatenate to a maximum of 256 characters. Examples: ◆ SSL_KEY_DBASE_PW=’My test password’ ◆ SSL_KEY_DBASE_PW=’My test password containing an embedded (‘) apostrophe’ ◆ SSL_KEY_DBASE_PW=’My test password containing multiple embedded (“”’) apostrophes’ ◆ SSL_KEY_DBASE_PW=’This is a test password’ SSL_KEY_DBASE_PW=’used to demonstrate how to’ SSL_KEY_DBASE_PW=’continue onto the next ODF record’ Resolves to: This is a test password used to demonstrate how to continue onto the next ODF record. ◆ SSL_KEY_DBASE_PW=’This is a test password’ SSL_KEY_DBASE_PW=’containing three “’ embedded apostrophes’ Resolves to: This is a test password containing three “’ embedded apostrophes

Parameter	Description
SSL_KEYRING_LABEL=’ring-label’	Required if SSL=YES and certificate is created using RACF. Specifies the RACF KEYRING label-name used by the RACDCERT ADD command when a certificate/private key is defined. **Note:** The SSL_KEYRING_LABEL=’ring-label’ and SSL_KEYRING_NAME=’ring-name’ parameters must be used together and are mutually exclusive of the SSL_KEY_DBASE=’key-data-base-path-name’, SSL_KEY_DBASE_PW=’key-data-base-password’, and SSL_SERVER_CERT=’certificate-label-string’ parameters. **Note:** An invalid SSL_KEYRING_LABEL will cause a failure during SSL/TLS sessions.
SSL_KEYRING_NAME=’ring-name’	Required if SSL=YES and certificate is created using RACF. Specifies the RACF KEYRING ring-name used by the RACDCERT CONNECT command when adding a certificate/private key to one or more existing RACF key rings. **Note:** The SSL_KEYRING_LABEL=’ring-label’ and SSL_KEYRING_NAME=’ring-name’ parameters must be used together and are mutually exclusive of the SSL_KEY_DBASE=’key-data-base-path-name’, SSL_KEY_DBASE_PW=’key-data-base-password’, and SSL_SERVER_CERT=’certificate-label-string’ parameters.

Parameter	Description
SSL_SERVER_CERT= ’certificate-label-string’	Required if SSL=YES. Specifies a character string that contains the label for the key in the key database file. This value retrieves the Connect:Enterprise server certificate. The character string is enclosed in single quotes and may contain embedded blanks, quotation marks, and any other characters. The outer single quotes are used as the string delimiters. The character string may continue over multiple ODF records by specifying multiple consecutive SSL_SERVER_CERT= parameters. The strings concatenate to a maximum of 256 characters. Examples: ◆ SSL_SERVER_CERT=’My test certificate’ ◆ SSL_SERVER_CERT=’My test certificate containing an embedded (‘) apostrophe’ ◆ SSL_SERVER_CERT=’My test certificate containing multiple embedded (“’’) apostrophes’ ◆ SSL_SERVER_CERT=’This is a test certificate label’ SSL_SERVER_CERT=’used to demonstrate how to’ SSL_SERVER_CERT=’continue onto the next ODF record’ Resolves to: This is a test certificate label used to demonstrate how to continue onto the next ODF record. ◆ SSL_SERVER_CERT=’This is a test certificate label’ SSL_SERVER_CERT=’containing three “’ embedded apostrophes’ Resolves to: This is a test certificate label containing three “’ embedded apostrophes
SSL_TIMEOUT=00300 | nnnnn	Specifies the number of seconds for the SSL session identifier to expire. The range is 0–86400 seconds (1 day). Specifies the maximum amount of time (in seconds) that System SSL retains the SSL V3 session identifiers. This reduces the amount of data exchange during the SSL handshake for peer applications when a complete initial handshake has already been performed within the allotted time.

Parameter	Description
SYST215='your desired text &OSNAME &OSVER'	Specifies the FTP server SYST 215 reply text for all FTP servers. To substitute the operating system name and version, use the &OSNAME and &OSVER variables. The default is: 215 MVS OSNAME OSVER is the operating system for Connect:Enterprise Vxx.Rxx.Mxx **Note:** To set the FTP Server SYST 215 reply text for a particular FTP server, add SYST215='your desired text &OSNAME &OSVER' to your ODF *REMOTE section. For more information, see page 143.

Configuring *REMOTES Records for FTP Connections

The *REMOTES record enables you to define some parameters different from the default values set in the *OPTIONS record for connections to and from remote FTP sites. Some FTP *OPTIONS parameters have counterparts in the *REMOTES record to enable you to define site-specific values that override the global settings. For example, you set the SSL_DEFAULT_POLICY parameter in the *OPTIONS record. This value defines the use of SSL for all FTP sessions unless you override it in the SSL_POLICY parameter in the FTP client and server *REMOTES record.

Each remote FTP site that can establish a session with the Connect:Enterprise for z/OS host must be defined as a client to Connect:Enterprise in the *REMOTES section of the ODF. Likewise, each FTP site that the Connect:Enterprise for z/OS host connects to must be defined as a server in the *REMOTES record. Values set in the *REMOTES record take precedence over the FTP values set in the *OPTIONS record for FTP client and server connections.

All remote sites defined to Connect:Enterprise have a remote name. For FTP, this remote name is supplied by user name to Connect:Enterprise for z/OS as the DATA portion of a logon to Connect:Enterprise. The remote name then accesses a table of information that is built from the *REMOTES records.

The *REMOTES record enables you to:

- Supply options unique to a remote site
- Provide default batch Mailbox IDs (remote name is used)
- Determine which batches to transmit during an Auto Connect session

*REMOTES Record Rules

When you configure the *REMOTES record, observe the following rules:

- *REMOTES must begin in column 1. Any other text on the same line after *REMOTES is ignored.
- The NAME= keyword is required and must be the first keyword of the record.

- The TYPE= keyword is required and must follow the NAME= keyword.
- Required keywords must precede the optional keywords.
- Keywords can begin in any column and can include multiple values.
- Optional keywords can be in any order.

*REMOTES Record Format for FTP Client Connections

The following sample illustrates the FTP client *REMOTES record format. This record defines default values for remote-initiated client communications sessions to the Connect:Enterprise for z/OS server.

```
*REMOTES
  NAME=xxxxxxxx | ANONYMOUS | [generic remote]*
  TYPE=FTP_CLIENT
  BCHSEP=NONE | OPT3 | OPT4
  DIR_FILTER=D | flags
  DIRFORM=BROWSER|BROWSER64|MBOX_CLIENT|MBOX_CLIENT64|MBOX_ZOS|MBOX_ZOS64|$MBINSDFXY|
     UNIX|UNIX64
  DISCINTV=0-3600
  EDI=YES|NO
  FTP_ALLOW_GETBYNBR_DFLAG=NO|YES
  FTP_DATA_PORT_RANGE=nnnnn-nnnnn, nnnnn-nnnnn | L-1
  FTP_PORT_RETRIES=1-99
  FTP_PORT_RETRY_WAIT_TIME=nnn
  KIRN=NO | YES
  LS_FILTER=BDI!RST | flags
  ONEBATCH=NO|YES
  PASSWORD_CASE=UPPER | MIXED | BOTH
  RECEIVE_OPTIONS= (BID=’NONE’ | ’<64 byte string>’, EO=NO|YES, MULTXMIT=NO|YES,
       RENAME=BID | FIRST24 | LAST24 | FIRST64 | LAST64, TO=NO|YES, XMIT=NO|YES,)
  REMOTE_FILENAME_LENGTH=SHORT | LONG | LONG64
  RIFS=YES | NO
  SCAN=NO |YES |ALL
  SSL_CCC_POLICY=DISALLOWED | REQUIRED | OPTIONAL
  SSL_CLIENT_AUTH_POLICY=REQUIRED | DISALLOWED | OPTIONAL
  SSL_POLICY=REQUIRED | DISALLOWED | OPTIONAL
  SYST215='your desired text &OSNAME &OSVER'
  TRANSLATE=translate table name | STANDARD
```

Generic FTP Remote Site Definitions

To log on to Connect:Enterprise, an FTP user needs an ODF *REMOTES definition that contains the remote name supplied by the user logon. However, you can avoid creating a *REMOTES definition for each FTP user by using generic *REMOTES definitions. Generic *REMOTES definitions group FTP client remote sites that share connection parameter values. Each remote in the group can continue to use its unique remote name without needing an individual *REMOTES definition.

Note: Be careful with generic remote definitions, because a change to a generic definition affects all remote sites that use it.

In a generic *REMOTES definition, the remote name parameter (NAME) consists of 1–7 characters followed by an asterisk. For example, for a group of remote sites named RMT1 through RMT99, specify NAME=RMT* in the *REMOTES definition.

When you specify NAME=remotename*, you must also specify the TYPE=FTP_CLIENT parameter in the definition. This definition is only valid for remote-initiated connections.

You can still use the Connect:Enterprise security interface and security exit with generic remote sites. The complete remote name supplied by the user during logon (RMT1 in the preceding example) is used for logon and security checks. A password is required. Session activity logging will include the complete remote name. Application agents are also invoked using the complete remote name.

Anonymous FTP Remote Site Definitions

Anonymous FTP remote sites can log on to Connect:Enterprise from remote FTP site and execute all FTP commands, except for the CWD and DELE commands. You can further restrict the list of commands available for the anonymous user through the Connect:Enterprise security interface or a security exit.

To set up an anonymous FTP remote, you must define an ODF *REMOTES record where the remote name is “ANONYMOUS” (NAME=ANONYMOUS). Define the remainder of the *REMOTES record as you would for any FTP remote.

After the *REMOTES record is defined, the user can log on to Connect:Enterprise by specifying “anonymous” as the USERID and supplying a password.

The Connect:Enterprise security interface does not perform logon checking on anonymous FTP. However, you can create a security exit to check the password if desired.

The default mailbox ID for anonymous remote sites is PUBLIC.

*REMOTES Record Parameters for FTP Client Connections

The following table lists *REMOTES parameters that define connections from remote FTP clients to the Connect:Enterprise for z/OS server. The required parameters are listed in bold first in the table; the remaining optional parameters are listed in alphabetical order. Defaults are underlined.

Parameter	Description
NAME=<remote_name> | ANONYMOUS | [generic_remote]*	Required. Specified by the FTP client as a parameter on the USER server command. Referred to as the remote name on all Connect:Enterprise reports and interfaces. ◆ remote name—Specifies the 1–8 character name of the remote client. ◆ ANONYMOUS—Specifies the remote name for anonymous remote clients. Alternate security for anonymous logins bypasses SAF logon security, but control passes to the Site Security Exit. Anonymous remote sites are restricted to the Mailbox ID “PUBLIC.” ANONYMOUS is not a valid Mailbox ID. ◆ [generic remote]*—Specifies 0–7 characters followed by an asterisk ‘*’ to indicate a generic remote. Generic remote entries group FTP client remote sites with similar connection parameter values. For example, a group of NAME=RMT1 through RMT99 could use the single remote definition NAME=RMT*. **Note:** The sequence of generic remote definitions in *REMOTES is significant. A remote logon will use the first matching entry in the list of remote definitions. Longer, more specific names are listed first. An entry like NAME=AB* must precede an entry of NAME=A*. If a remote named ABC logs on and the shorter, more generic name is listed first in the ODF, ABC would be the first match. *The longer, more specific entry will only be used if it precedes the shorter entry.*
TYPE=FTP_CLIENT	Required. TYPE=FTP_CLIENT indicates this is an FTP remote client. This parameter must immediately follow NAME=.

Parameter	Description
BCHSEP=NONE | OPT3 | OPT4	For remote FTP clients retrieving files from Connect:Enterprise, the BCHSEP parameter indicates how multiple batches are concatenated and when they are flagged as transmitted. The BCHSEP value also affects the NLST response to list files either by BID or by batch number. Specification of this parameter defines the default value for each session. The remote FTP client using a SITE command may override the value of this parameter. ◆ NONE (default)—Specifies that multiple requested batches selected for processing, are concatenated and sent as a single file. Each batch is flagged transmitted after its last record is sent. A subsequent transmission failure after a batch is sent does not reset its transmitted flag and the batch is not available for retransmission without operator intervention. With BCHSEP=NONE, an NLST response includes all User Batch IDs with the current working Mailbox ID. If multiple batches have the same BID, a single entry is returned for the NLST representing one file that may be requested containing concatenated batches. ◆ OPT3—Specifies that multiple batches are concatenated and sent as a single file. All the batches are flagged transmitted after the last batch is successfully sent. A transmission failure after a batch is sent will prevent the transmitted flag from being set, so all batches are then available for re-transmission. The NLST response is the same as with BCHSEP=NONE, listing all BIDs for the current Mailbox ID. ◆ OPT4—Specifies that the response to an NLST will include an entry for every batch in the current Mailbox ID. The format of the batch entries in the NLST is determined by the REMOTE_FILENAME_LENGTH parameter. The response to RETR command concatenates batches and sets transmitted flags the same as BCHSEP=NONE. Any RETR for a specific, generic or wildcard BID will cause the transfer of a single file containing all batches matching the current Mailbox ID and the BID. Each batch is flagged transmitted after its last record is sent.
DIR_FILTER=D | flags	Used by the execution of the LIST command as a filter to exclude batches from selection for the list returned to the remote FTP client. This parameter defines the default value for each session. A remote FTP client may override the value using a SITE command. See the *Connect:Enterprise for z/OS Remote User's Guide* for a listing of the flag values.

Parameter	Description
DIRFORM=BROWSER|BROWSER64 |MBOX_CLIENT|MBOX_CLIENT64| MBOX_ZOS|MBOX_ZOS64| $MIBNSDFXY|UNIX|UNIX64	Specifies the format of a line returned to the remote FTP client in response to the FTP server LIST command. This parameter defines the default value for each session. A remote FTP client can override the value using a SITE command. BROWSER = Specifies a format supported by browsers, displaying the first 24 characters of the Batch ID. BROWSER64 = Specifies a format supported by browsers, displaying the full 64 character Batch ID. MBOX_CLIENT = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the first 24 characters of the Batch ID. MBOX_CLIENT64 = Specifies a format supported by Connect:Enterprise Client for Windows and the Connect:Enterprise Command Line Client, displaying the full 64 character Batch ID. MBOX_ZOS = Specifies the Connect:Enterprise $$DIR format, displaying the first 24 characters of the Batch ID. MBOX_ZOS64 = Specifies the Connect:Enterprise $$DIR format, displaying the full 64 character Batch ID. $MBINSDFXY = Reply format options. You can specify as many options as you want and in any order after the initial $ option. ◆ $ = User-defined format ◆ M = Eight-character character Mailbox ID ◆ B = 24-character Batch ID (BID=xxxx….xxxx) ◆ I = 24-character Batch ID (xxxx….xxxx) ◆ N = Seven-digit batch number (#nnnnnn) ◆ S = Eight-digit file size in number of bytes (CT=nnnnnnnn) ◆ D = Time/date of batch creation (hhmm-yyddd) ◆ F = Batch status flags ◆ X = 64-character Batch ID (BID=xxxx….xxxx) ◆ Y = 64-character Batch ID (xxxx….xxxx) UNIX = Specifies the standard UNIX directory display format, displaying the first 24 characters of the Batch ID. UNIX64 = Specifies the standard UNIX directory display format, displaying the full 64 character Batch ID
DISCINTV=0-3600	Specifies the amount of time (in seconds) that a session may be inactive before forced termination occurs. If no value is specified, the value of the FTP_DEFAULT_DISCINTV *OPTIONS parameter is used. ◆ 0— Does not force a disconnect. ◆ 1-3600—Connect:Enterprise forces a session end if there is no session activity for the number of seconds specified.

Parameter	Description
EDI=YES|NO	Specifies whether single byte X'15' segment terminators are used. ◆ YES—Indicates X'15' segment terminators are being used and allows the translation table to translate the X '15' to a single byte. ◆ NO—Indicates X'15' segment terminators are not being used so standard EBCDIC to ASCII translation is used to translate the X '15' to the 2-byte X '0D0A'.
FTP_ALLOW_GETBYNBR_DFLAG= NO|YES	Specifies whether remote clients are allowed to retrieve batches from this remote site, by batch number, even if the selected batch has been marked deleted. If this parameter is not specified, the value from FTP_ALLOW_GETBYNBR_DFLAG_DEFAULT in the ODF *OPTIONS is used for this remote. ◆ NO—Do not allow ◆ YES—Do allow
FTP_DATA_PORT_ RANGE=L-1 | nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of ports the Connect:Enterprise FTP server uses to transfer data to a remote FTP client. The default is defined by the value set in the FTP_DEFAULT_SERVER_DATA_PORT_RANGE parameter in the *OPTIONS section of the ODF. ◆ L-1—A special value that sets the data port to the logon listen port number minus one. Used when the server connects back to a known port number on the client. ◆ nnnnn-nnnnn—Specifies a range of port numbers. Ranges contain the lowest to the highest port numbers available in that range. Separate ranges with commas. A single port is designated by setting the same value in both the low and high port number fields.
FTP_PORT_RETRIES=nn	Specifies how many times (from 0–99) a connection attempt is made for each port in the defined range or ranges. The default value is defined by the value set in FTP_DEFAULT_PORT_RETRIES in the *OPTIONS section of the ODF.
FTP__RETRY_WAIT_TIME=nnn	Specifies the number of seconds (from 0–180) the server waits between connection attempts. The default value is defined by the value set in FTP_DEFAULT_RETRY_WAIT_TIME in the *OPTIONS section of the ODF.

Parameter	Description
KIRN=YES | NO	(KIRN stands for Keep Input Recsep NL) Specifies whether or not Connect:Enterprise removes the record separator string when the batch is stored. The default comes from the FTP_DEFAULT_KIRN setting in the *OPTIONS record, which is NO. NO = Connect:Enterprise for z/OS removes the record separator string after recordizing the batch. YES = The Record separator strings are kept in the batch. The corresponding FTP_DEFAULT_RIFS parameter must be set to YES. See *Processing $$ADD Commands Embedded in Batches* on page 103 for more information.
LS_FILTER=BDI!RST | flags	Used by the NLST command to exclude batches from selection for the list returned to the remote FTP client. An exclamation mark preceding a flag filters out batches that have that status flag. The default value, LS_FILTER=BDI!RST, removes all BSC collected, deleted batches, incomplete batches, previously transmitted batches, SNA collected and nonrequestable batches from the returned directory list. This parameter defines the default value for each session. A remote FTP client may override the value using a SITE command.
ONEBATCH=NO|YES	Specifies whether only the first batch that meets the transmission criteria be sent. - YES—Specifies that only the first eligible batch is selected for transfer to the remote FTP client. - NO—Specifies that all eligible batches are selected for transfer to the remote FTP client. This parameter defines the default value for each session. A remote FTP client may override the value using a SITE command.

Parameter	Description
PASSWORD_CASE=UPPER | MIXED | BOTH	Specifies how passwords are presented to the security package at logon authorization, in terms of case sensitivity. ◆ UPPER—The password is uppercased before it is presented to the security package for logon security authorization. Only the upper cased value is validated by the security package. For example, if the user enters a password value of "MyPass", the value is uppercased to "MYPASS". If a new password is provided in the logon request, it is also uppercased. ◆ MIXED—The password is not uppercased before it is presented to the security package for logon security authorization. Only the original password value, as entered by the user, is validated by the security package. ◆ BOTH—Both mixed and uppercase password values are validated by the security package, if necessary. First, the original value (mixed case) is validated by the security package. If the logon check fails using the mixed case password, a second attempt is made to validate the password using the uppercased value. A successful logon check with either mixed or upper cased password is considered valid. The intended purpose of specifying BOTH is to allow a transition period for some duration after your systems programmer / security administrator turns on mixed case password support in the security package. This enables Connect:Enterprise to successfully allow users/remotes to logon. Eventually, after users have changed the password with mixed case turned on in the security package, you should set PASSWORD_CASE=MIXED. **Note:** When BOTH is specified, if the first attempt fails (mixed case), but the second attempt is successful (uppercase), Connect:Enterprise considers the logon successful and continues processing as normal. However, the security package still posts a security violation console message due to the failure of the first logon check attempt in mixed case. Also be aware that if both attempts fail, the "consecutive unsuccessful password attempts" count, maintained within the security package, is incremented by 2, since two individual calls are made to the security package.

Parameter	Description
RECEIVE_OPTIONS|R_O= (BID='NONE' | '<64 byte string>', EO=NO|YES, MULTXMIT=NO|YES, RENAME=BID24 | FIRST24 | LAST24 | BID64 | FIRST64 | LAST64, TO=NO|YES, XMIT=NO|YES,)	Specifies the default values for BID, EO, MULTXMIT, RENAME, TO, and XMIT. These values are used to create batches that are sent to Connect:Enterprise from the remote FTP client using the STOR or STOU command. The remote FTP client can use a SITE command to override the value of this parameter. The R_O= parameter can be spread across multiple 80-byte lines as long any single sub-parm is completely on the line. For example, this is valid: RECEIVE_OPTIONS=(BID='This is a short bid', RENAME=FIRST24,XMIT=NO) This is not valid (sub-parm BID split): RECEIVE_OPTIONS=(BID='The start of my bid',EO=YES,XMIT=NO) The BID value is assigned to any batch received in a STOU transfer from the remote server. The batch ID assigned by the BID value uniquely identifies each batch. **Note:** The default value for BID is 'NONE'. EO—Specifies the extract once option for batches being received. MULTXMIT—Specifies the multiple transmission flag for batches being received. The RENAME value provides options to create a unique batch ID when file names in a STOR command exceed 24 or 64 characters: ◆ BID24 replaces any STOR file name that exceeds 24 characters with the BID value. ◆ First24—Truncates a long file name by using the first 24 characters (including blanks) as the batch ID (default value). ◆ Last24—Truncates a long file name by using the last 24 characters (including non-trailing blanks) as the batch ID. Suffixes, such as .TXT, are included. ◆ BID64 replaces any STOR file name that exceeds 64 characters with the BID value. ◆ First64 truncates a long file name by using the first 64 characters of the inbound file name as the User Batch ID. ◆ Last64—Truncates a long file name by using the last 64 characters of the inbound file name, as the User Batch ID. TO—Specifies the transmission control limits for the batch being received. XMIT—Specifies the transmission control limits for the batch being received.

Parameter	Description
REMOTE_FILENAME_ LENGTH = LONG | SHORT | LONG64	Specifies the format of the file name created by the Connect:Enterprise FTP server when BCHSEP=OPT4 is specified. LONG = Uses the 24 character User Batch ID as the filename format. SHORT = Uses the 7-character batch number as the filename format. LONG64 = Uses the 64 character user batch ID as the filename format
RIFS=YES | NO	(RIFS stands for Recordize Input File Structure) Changes the batch to record structure or retains the batch as file structure. YES = Recordizes the batch after recognizing a record separator string and uses CRLF (x'0D0A) for SFA batches and NL (x'15') for SFE batches. NO = Retains file structure of batch and does not recognize record separator strings in SFA or SFE batches. The default comes from the FTP_DEFAULT_RIFS setting in the *OPTIONS record, which is YES. See *Recordizing Batches* on page 113 for more information. **Note:** Processing results cannot be predicted or supported when RIFS=NO and SCAN is set to YES or ALL.
SCAN=NO |YES |ALL	Specifies whether for this remote site Connect:Enterprise for z/OS scans received batches for $$ commands. NO= Scanning for Connect:Enterprise for z/OS $$ commands is not enabled. Connect:Enterprise for z/OS $$ commands, /*SIGNON, and /*BINASC cards embedded in a received batch are treated as data. YES= Scanning for Connect:Enterprise for z/OS $$ commands is enabled initially. To keep scanning in effect when a $$ADD card is encountered, each $$ADD card must include the parameter SCAN=YES. Use this value to make FTP command scanning behave like it does in SNA. ALL= Scanning for Connect:Enterprise for z/OS $$ commands is enabled for the entire batch unless the batch contains a $$ADD card with the parameter SCAN=NO. Use this value to make FTP command scanning behave like it does in BSC.
SSL_CCC_POLICY= DISALLOWED | REQUIRED | OPTIONAL	Sets the SSL_CCC_POLICY for a specific remote client definition. Overrides the SSL_DEFAULT_CLIENT_CCC_POLICY. ◆ DISALLOWED—The CCC command is not honored and the control session remains encrypted. This is the default value. ◆ OPTIONAL—Connect:Enterprise honors the CCC command if received from the client. No error results if the client does not send the CCC command. ◆ REQUIRED—Connect:Enterprise must process the CCC command before any data port operation can be attempted

Parameter	Description
SSL_CLIENT_AUTH_POLICY= REQUIRED | DISALLOWED | OPTIONAL	- REQUIRED—Specifies that connections between the remote site and Connect:Enterprise *must* be made secure using SSL. When REQUIRED is set, the Connect:Enterprise for z/OS FTP server requests the client certificate and authenticates it. - DISALLOWED—Specifies that connections between the remote site and Connect:Enterprise *will not* be made secure using SSL. When DISALLOWED is set, the Connect:Enterprise for z/OS FTP server does not request the client certificate. - OPTIONAL—Specifies that connections between the remote site and Connect:Enterprise *can* be made secure using SSL at the client's discretion. When OPTIONAL is set, the Connect:Enterprise for z/OS FTP server requests the client certificate. If no client certificate is available, the session continues successfully. If a client certificate is available, it will be sent to the server for client authentication. If the client authentication fails, the session fails. Therefore, if you do not want the Connect:Enterprise for z/OS FTP server to attempt to perform client authentication, set DISALLOWED.
SSL_POLICY=OPTIONAL| REQUIRED | DISALLOWED	Sets the SSL requirement between the remote client and the server when Connect:Enterprise is acting as an FTP server. Overrides the SSL_DEFAULT_POLICY set in the *OPTIONS section of the ODF. - OPTIONAL—Specifies that connections between the remote site and Connect:Enterprise *can* be made secure using SSL or TLS at the client's discretion. If AUTH SSL or TLS is received, the control channel will be secure. Unless PBSZ:PROT (Protection Buffer Size:Data Channel Protection Level) is received to change PROT (Protection Level) from the default value of C (Clear), data on the data channel will be in the clear. - REQUIRED—Specifies that connections between the remote site and Connect:Enterprise *must* be made secure using SSL or TLS. As client, Connect:Enterprise automatically sends PBSZ 0 and PROT P (Private) to secure both the control and data channels. As server, Connect:Enterprise automatically sets PBSZ 0 and PROT P on both the control and data channels even if these commands are not received from the client . A non-Connect:Enterprise client may still explicitly send the PBSZ and PROT P commands. - DISALLOWED—Specifies that connections between the remote site and Connect:Enterprise will not be made secure using SSL or TLS. Both the control channel and data channel will be in the clear. See *Setting Up Support for SSL Protocol* on page 102 for more information.

Parameter	Description
SYST215='your desired text &OSNAME &OSVER'	Specifies the FTP server SYST 215 reply text for this FTP server. To substitute the operating system name and version, use the &OSNAME and &OSVER variables. The default is: 215 MVS OSNAME OSVER is the operating system for Connect:Enterprise Vxx.Rxx.Mxx **Note:** To set the FTP Server SYST 215 reply text on a global basis to use this text for all FTP servers, add SYST215='your desired text &OSNAME &OSVER' to your ODF *OPTIONS section. For more information, see page 131.
TRANSLATE=translate table name | <u>STANDARD</u>	Specifies the name of the translation table to use when converting ASCII data to EBCDIC data or EBCDIC data to ASCII data. - translate-table-name—Enables you to specify a customized table. If you wish to use a custom translation table, you create one using the IBM CONVXLAT utility. Refer to IBM documentation for how to create custom translation tables. Once created, each table must be stored as a PDS member in the TRANSLAT DD data set specified in the Connect:Enterprise startup JCL. An example member is supplied in ENTPRS.EXAMPLE(CONVXLAT). Translation occurs for all nontransparent batches sent or received in ASCII format. All nontransparent batches in the VSAM Batch Queues are in EBCDIC character representation.

*REMOTES Record Format for FTP Server Connections

The following sample format is for the FTP server *REMOTES records. This record defines default values for Connect:Enterprise for z/OS client connections to remote FTP servers.

```
*REMOTES
NAME=xxxxxxxx
  TYPE=FTP_SERVER
  &IPADDR=hostname
  &BID=’NONE’|’xxx...xxx’
  &DATAMODE=B|C|S
  &PASSWORD=xxxxxx...xxx
  &NEWPASS=xxxxxx...xxx
  &PORTNO=21|nnnn
  &RECVPATH=directory_path
  &SENDPATH=directory_path
  &DATASTRU=F|R
  &DATATYPE=A|E|I
  &USERID=remote_name|xxxxxxxx
  BCHSEP=NONE|OPT3|OPT4
  DISCINTV=0-3600
  EDI=YES|NO
  IDENT=NO|YES
  KIRN=NO | YES
  LOGON_SCRIPT=xxxxxxxx
  FTP_CONTROL_PORT_RANGE=nnnnn-nnnnn
  FTP_DATA_PORT_RANGE=nnnnn-nnnnn | U
  FTP_PORT_RETRIES=0 | nn
  FTP_PORT_RETRY_WAIT_TIME=030 | nnn
  REMOTE_FILENAME_LENGTH=SHORT | LONG | LONG64
  RIFS=YES | NO
  SCAN=NO |YES |ALL
  SENDPASV=NO|YES
  SENDSITE=NO|YES
  SSL_CCC_POLICY=REQUIRED|DISALLOWED|OPTIONAL
  SSL_POLICY=REQUIRED|DISALLOWED|OPTIONAL
  TRANSLATE=pds_member_name|STANDARD
```

*REMOTES Record Parameters for FTP Server Connections

The following table lists the *REMOTES parameters that define Connect:Enterprise for z/OS client connections to remote FTP servers. Required parameters are listed in bold first in the table; the remaining optional parameters are listed in alphabetical order. Defaults are underlined.

Parameter	Description
NAME=xxxxxxxx	Required. Specifies the 1–8 character name of the remote FTP server.
TYPE=FTP_SERVER	Required. Identifies an FTP remote server. This parameter must immediately follow NAME=.
&BID=’NONE’|’xxx...xxx’	Sets the value of the 1–64 character BID variable used in the AC_SCRIPT. The value must be enclosed in single quotes. Embedded blanks are permitted. If not specified, the default is NONE.

Parameter	Description
&DATAMODE=B|C|S	Sets the value of the MODE variable used in the AC_SCRIPT. ◆ S—Stream ◆ B—Blocked ◆ C—Compressed
&IPADDR=hostname	Sets the value of the IPADDR variable used in the LOGON_SCRIPT. The value must be in the form of host name (or IP address). The maximum length of the host name value is 60 characters.
&NEWPASS=xxxxxx...xxx	Sets the case sensitive value of the 1–64 character NEWPASS variable used in the AC_SCRIPT. Embedded blanks are not permitted. The &NEWPASS value is masked when printing to SYSPRINT.
&PASSWORD=xxxxxx...xxx	Sets the case sensitive value of the 1–64 character PASSWORD variable used in the AC_SCRIPT. Embedded blanks are not permitted. The &PASSWORD value is masked when printing to SYSPRINT.
&PORTNO=21|nnnn	Sets the port number to use when connecting to the remote server. The default is 21.
&RECVPATH=‘directory_path’	Sets the value of the 1–64 character, case-sensitive RECVPATH variable used in the AC_SCRIPT. Embedded blanks enclosed in single quotes are permitted.
&SENDPATH=‘directory_path’	Sets the value of the 1–64 character, case-sensitive SENDPATH variable used in the AC_SCRIPT. Embedded blanks enclosed in single quotes are permitted.
&DATASTRU=F|R	Sets the value of the DATASTRU variable used in the AC_SCRIPT. ◆ F—File ◆ R—Record
&DATATYPE=A|E|I	Sets the value of the DATATYPE variable used in the AC_SCRIPT. ◆ A—ASCII ◆ E—EBCDIC ◆ I—Image **Note:** Connect:Enterprise automatically converts TYPE=I to Binary before routing the data to the server.
&USERID=remote_name|xxxx xxxx	Sets the value of the 1–10 character, case-sensitive USERID stem variable used in the LOGON_SCRIPT. Embedded blanks are not permitted. If not specified, the default is the remote name specified in the NAME parameter of the remote definition.

Parameter	Description
BCHSEP=NONE | OPT3 | OPT4	Specifies how Connect:Enterprise processes batches sent to a remote site. All options of BCHSEP apply to Auto Connect session STOR or STOU commands that transmit multiple batches. The BCHSEP value may be overridden for a given session using an AC_SCRIPT. ◆ NONE (default)—Concatenates all batches and sends as a single file. Each batch is flagged transmitted after its last record is sent. A subsequent transmission failure after a batch is sent does not reset its transmitted flag and the batch is not available for retransmission without operator intervention. ◆ OPT3—Concatenates multiple files and sends as a single file. All the batches are flagged transmitted after the last batch is successfully sent. A transmission failure at any point prevents the transmitted flag from being set, so all batches are then available for re-transmission. A failure of a STOR leaves all batches in that file available for retransmission, even if some batches were successfully sent within the same STOR before the error. ◆ OPT4—Sends each eligible batch as a single file. Each batch is flagged transmitted after its last record is sent.
DISCINTV=0-3600	Specifies the amount of time (in seconds) that a session may be inactive before a forced termination occurs. If no value is specified, the value of the FTP_DEFAULT_DISCINTV *OPTIONS parameter is used. ◆ 0— Does not force a disconnect. ◆ 1-3600—Connect:Enterprise forces a session end if there is no session activity for the number of seconds specified.
EDI=YES|NO	Specifies whether single byte X’15’ segment terminators are used. ◆ YES—Indicates X’15’ segment terminators are being used and allows the translation table to translate the X '15' to a single byte. ◆ NO—(Default) Indicates X’15’ segment terminators are not being used so standard EBCDIC to ASCII translation is used to translate the X '15' to the 2-byte X '0D0A'.
IDENT=YES|NO	Indicates whether Connect:Enterprise attempts to determine whether the FTP server is another Connect:Enterprise product. ◆ YES—Connect:Enterprise attempts to determine whether the FTP server is another Connect:Enterprise product. ◆ NO—Connect:Enterprise does not attempt to determine whether the FTP server is another Connect:Enterprise product. All processing assumes the remote FTP server is not a Connect:Enterprise product. **Note:** When the remote server is Connect:Enterprise Gateway or Connect:Enterprise for UNIX, specify IDENT=YES.

Parameter	Description
KIRN=YES | NO	(KIRN stands for Keep Input Recsep NL) Connect:Enterprise for z/OS removes the record separator string so that the batch is stored as a file structure instead of being record-oriented or keeps the record separator string, NL (New Line feed), for incoming SFA and SFE batches or allows. If the batch is not recordized, this parameter is ignored. The default comes from the FTP_DEFAULT_KIRN setting in the *OPTIONS record, which is NO. NO = Connect:Enterprise for z/OS removes the record separator string after recordizing the batch. YES = The Record separator strings are kept in the batch. The corresponding FTP_DEFAULT_RIFS parameter must be set to YES. See *Recordizing Batches* on page 113 for more information.
LOGON_SCRIPT=xxxxxxxx	Specifies the member name of the LOGON_SCRIPT used to log on to the remote server and negotiate firewalls. The LOGON_SCRIPT must be a PDS member in a file allocated to DD SYSEXEC in the Connect:Enterprise JCL. If no value is specified, the default is determined by the setting of the FTP_LOGON_SCRIPT_DEFAULT parameter in the *OPTIONS section of the ODF.
FTP_DATA_PORT_ RANGE=U | nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of data ports a Connect:Enterprise FTP client uses to transfer data to an FTP server. Ranges contain the lowest to the highest port number available in that range. Separate ranges with commas. The default is specified in FTP_DEFAULT_CLIENT_DATA_PORT_RANGE in the *OPTIONS section of the ODF. ◆ U—Sets the auto connect client data port number to re-use the client control port number used to logon.
FTP_CONTROL_PORT_ RANGE=nnnnn-nnnnn, nnnnn-nnnnn	Specifies up to five ranges of control ports a Connect:Enterprise FTP client uses to connect to an FTP server. Ranges contain the lowest to the highest port number available in that range. Separate ranges with commas. The default is specified in FTP_DEFAULT_CLIENT_CONTROL_PORT_RANGE in the *OPTIONS section of the ODF.
FTP_PORT_RETRIES=nn	Specifies how many times (from 0–99) a connection attempt is made for each data port or control port in the defined range or ranges. The default value is defined by the value set in FTP_DEFAULT_PORT_RETRIES in the *OPTIONS section of the ODF.
FTP_PORT_RETRY_WAIT_ TIME=nnn	Specifies the number of seconds (from 0–180) the server waits between connection attempts to a control port or a data port. The default value is defined by the value set in FTP_DEFAULT_PORT_RETRY_WAIT_TIME in the *OPTIONS section of the ODF.

Parameter	Description
REMOTE_FILENAME_ LENGTH =LONG | SHORT | LONG64	Specifies the format of the file name created by the Connect:Enterprise FTP server when sending data to the remote FTP server when using the STOR command. This parameter defines the default value for each session. You can change the value of this parameter within an Auto Connect script using the locsite command. LONG = Uses the 24 character User Batch ID as the filename format. SHORT = Uses the seven-character batch number as the filename format. LONG64 = Uses the 64 batch User ID as the filename format
RIFS=YES | NO	(RIFS stands for Recordize Input File Structure) Changes the batch to record structure or retains the batch as file structure. YES = Recordizes the batch after recognizing a record separator string and uses CRLF for SFA batches and NL for SFE batches. NO = Retains file structure of batch and does not recognize record separator strings in SFA or SFE batches. The default comes from the FTP_DEFAULT_RIFS setting in the *OPTIONS record, which is YES. See *Recordizing Batches* on page 113 for more information. **Note:** Processing results cannot be predicted or supported when RIFS=NO and SCAN is set to YES or ALL.
SCAN=NO |YES |ALL	Specifies whether for this remote site Connect:Enterprise for z/OS scans received batches for $$ commands. NO = Scanning for Connect:Enterprise for z/OS $$ commands is not enabled. Connect:Enterprise for z/OS $$ commands, /*SIGNON, and /*BINASC cards embedded in a received batch are treated as data. YES = Scanning for Connect:Enterprise for z/OS $$ commands is enabled initially. To keep scanning in effect when a $$ADD card is encountered, each $$ADD card must include the parameter SCAN=YES. Use this value to make FTP command scanning behave like it does in SNA. ALL= Scanning for Connect:Enterprise for z/OS $$ commands is enabled for the entire batch unless the batch contains a $$ADD card with the parameter SCAN=NO. Use this value to make FTP command scanning behave like it does in BSC.
SENDPASV=NO|YES	Specifies whether Connect:Enterprise sends the PASV or PORT command to the remote FTP Server to open a data connection. ◆ NO—Specifies that a PORT command be used to open a data connection with the remote FTP Server. ◆ YES—Specifies that a PASV command be used to open a data connection with the remote FTP Server. **Note:** Yes is usually required when transferring through a firewall.

Parameter	**Description**
SENDSITE=NO|YES	Specifies whether Connect:Enterprise sends a SITE command to identify the physical characteristics of the file prior to issuing the STOR or STOU command. ◆ NO—Specifies that a SITE command is not issued automatically. A specific SITE command can still be included in the script. ◆ YES—Specifies that a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx command be issued prior to issuing the STOR or STOU command. The values of LRECL, BLKSIZE and RECFM are stored for the batch. If no values are available, the SITE command is not used.
SSL_CCC_POLICY= DISALLOWED | REQUIRED | OPTIONAL	Specifies the SSL_CCC_POLICY for a remote server definition. Overrides the SSL_DEFAULT_SERVER_CCC_POLICY set in the *OPTIONS section of the ODF. ◆ DISALLOWED—The CCC command is not issued by Connect:Enterprise. This is the default value. ◆ REQUIRED—The CCC command is issued. If the remote server rejects the command a QUIT command is issued. ◆ OPTIONAL—The CCC command is issued and, if accepted, it is honored. If it is rejected, the session remains active and the control connection is encrypted.
SSL_POLICY=OPTIONAL| REQUIRED | DISALLOWED	Sets the SSL requirement between the remote server and the client when Connect:Enterprise is acting as an FTP client. Overrides the SSL_DEFAULT_POLICY set in the *OPTIONS section of the ODF. ◆ OPTIONAL—Specifies that connections between the remote site and Connect:Enterprise *can* be made secure using SSL or TLS at the client's discretion. If AUTH SSL or TLS is received, the control channel will be secure. Unless PBSZ:PROT (Protection Buffer Size:Data Channel Protection Level) is received to change PROT (Protection Level) from the default value of C (Clear), data on the data channel will be in the clear. ◆ REQUIRED—Specifies that connections between the remote site and Connect:Enterprise *must* be made secure using SSL or TLS. As client, Connect:Enterprise automatically sends PBSZ 0 and PROT P (Private) to secure both the control and data channels. As server, Connect:Enterprise automatically sets PBSZ 0 and PROT P on both the control and data channels even if these commands are not received from the client. A non-Connect:Enterprise client may still explicitly send the PBSZ and PROT P commands. ◆ DISALLOWED—Specifies that connections between the remote site and Connect:Enterprise will not be made secure using SSL or TLS. Both the control channel and data channel will be in the clear. **Note:** See the *Connect:Enterprise for z/OS User's Guide* for more information and configuration examples.
TRANSLATE=pds_member_ name| STANDARD	Specifies the name of the translation table to use when converting ASCII data to EBCDIC data or EBCDIC data to ASCII data. A standard conversion is the default. If a member name is specified, it must exist in the file specified in the TRANSLAT DD.

Defining Custom Translation Tables for Remote Sites

Connect:Enterprise uses the standard ASCII-to-EBCDIC translation table provided with IBM TCP/IP. If you want to use the standard table you can skip this procedure.

If you want to use a custom translation table, use the following procedure:

1. Use the IBM CONVXLAT utility to create a custom translation table. See the IBM documentation for more information on creating customized translation tables.

   A sample CONVXLAT member is also provided in the EXAMPLE library.
2. Copy the table to the data set specified in the //TRANSLAT DD statement in the Connect:Enterprise JCL. This file is a PDS with the characteristics BLKSIZE=256, RECFM=F.
3. Use the TRANSLATE parameter in the *REMOTES ODF record to specify the name of the customer translation table. Use a unique member name when creating custom translation tables.

Connect:Enterprise supports only single-byte character set translation tables.

Sample FTP *REMOTES Records

The following sample *REMOTES record illustrates parameters for five FTP sites, including an Anonymous FTP remote client definition:

```
*REMOTES  DEFINE FIVE REMOTE SITES (5 FTP) WITH A VARIETY OF OPTIONS
**---------------FTP Remote Client Definitions--------------------------------------
  NAME=RMT001
    TYPE=FTP_CLIENT
    DIR_FILTER=DT
    ONEBATCH=YES
  NAME=RMT002
    TYPE=FTP_CLIENT
    RECEIVE_OPTIONS=(MULTXMIT=YES)
  NAME=RMT003*
    TYPE=FTP_CLIENT
    SSL_POLICY=REQUIRED
  NAME=ANONYMOUS
    TYPE=FTP_CLIENT
**---------------FTP Remote Server Definitions--------------------------------------
NAME=RMT005
    TYPE=FTP_SERVER
    LOGON_SCRIPT=LG005
    &PORT=5565
    &USERID=RMT5USR
    &PASSWORD=MYSEC
```

- Remote device 1 is an FTP client. Whenever a LIST command is received, any batch with a status flag of ‘D’ (deleted) or ‘T’ (transmitted) will be filtered out and will NOT be presented in the LIST reply. Only the first eligible batch selected for transfer will be sent to the remote.
- Remote device 2 is an FTP client. By default all batches collected from this remote are flagged as ‘M’ (multi-transmittable).

✦ Remote device 3 is a generic FTP client definition. Any remote user beginning with the characters ‘RMT003*’ will use this definition. All RMT003xx users are required to establish a secure connection using SSL.

✦ Remote device 4 specifies a remote name for ANONYMOUS FTP clients. Anonymous remote sites have restricted access to the Mailbox ID “PUBLIC”.

✦ Remote device 5 is an FTP server definition. When an Auto Connect session that specifies RMT005 starts, logon REXX script LG005 runs. REXX script LG005 can use variables &PORT, &USERID, and &PASSWORD to establish the FTP session level.

The following example illustrates a *REMOTES record for an FTP client that requires using SSL to secure the connection.

```
*REMOTES
    NAME=FTPRMT03
    DISCINTV=040
     BCHSEP=OPT4
     TYPE=FTP_CLIENT
     DIRFORM=$N
     DIR_FILTER=I
     REMOTE_FILENAME_LENGTH=LONG
     SSL_POLICY=REQUIRED
```

The following example illustrates a *REMOTES record for an FTP server that specifies to identify if the other server is a Connect:Enterprise for z/OS product.

```
*REMOTES
      NAME=FTPSRV1
           TYPE=FTP_SERVER
           BCHSEP=OPT4
           LOGON_SCRIPT=SRV1LGN
           IDENT=YES
          &IPADDR=MVSA
          &PORTNO=5575
          &USERID=MBOX1
          &PASSWORD=BANANA
          &DATASTRU=F
```

The following sample illustrates defining SSL parameters in the *REMOTES record.

```
*REMOTES
   NAME=FTPRMT03
      DISCINTV=040
      BCHSEP=OPT4
      TYPE=FTP_CLIENT
      DIRFORM=$N
      DIR_FILTER=I
      REMOTE_FILENAME_LENGTH=LONG
      SSL_POLICY=REQUIRED
      SSL_CLIENT_AUTH_POLICY=REQUIRED
  NAME=FTPRMT04
      TYPE=FTP_SERVER
      SSL_POLICY=REQUIRED
```

If the Client sends AUTH, followed by PBSZ and PROT P, both the control and data channels are encrypted. If the Client sends AUTH only (no subsequent PBSZ and PROT), Connect:Enterprise for z/OS uses “Implicit Data Channel Protection” and enforces encryption on both the control and data channels. If the Client sends AUTH, followed by PBSZ and PROT C, Connect:Enterprise for z/OS responds with reply code 534, which indicates that the server is not willing to accept the specified protection level.

About Auto Connect Sessions

An Auto Connect initiates the connection between the host site and the remote site. An Auto Connect session is either fully automated or manually initiated. Both data transmission and data collection can be performed during an Auto Connect session.

FTP Auto Connect sessions provide an interface with a remote FTP server implemented on any platform. FTP Auto Connects differ from SNA and BSC Auto Connects in that FTP Auto Connects use REXX language scripts to control both the connection to the remote server and data transmission between the client and remote server. These scripts execute automatically. They can be passed variables, which allow the scripts to be reused for different sessions. FTP Auto Connects use two scripts: the LOGON_SCRIPT and the AC_SCRIPT. These scripts are members in PDS files on the SYSEXEC DD. For a complete discussion of using REXX language scripts and variables for FTP Auto Connects, see Chapter 9, *FTP Auto Connect Scripts*.

An Auto Connect Manager (ACM) is responsible for the Auto Connect session, and ACM tasks can be replicated to allow for processing multiple concurrent requests. The ACM does not attempt retries which are accomplished using an AC_SCRIPT for FTP Auto Connects.

Fully automated Auto Connect sessions are activated each day when the system clock reaches the time of day specified in an Auto Connect list. If Connect:Enterprise remains up for multiple days, the Auto Connect session is activated every day when the system clock reaches the specified time. You can also define *CALENDAR records, and refer to them in the *CONNECT record, to specify dates and days of the week on which to activate or deactivate Auto Connect processing. See Chapter 7, *Configuring *CALENDAR Records*, for details.

A fully automated Auto Connect session is initiated by a date, day, or time specified in:

- *CONNECT record in the ODF
- User-written CICS API program

After it is set up, a fully automated Auto Connect session does not need operator intervention at the host site or the remote site, if the hardware at both sites can operate unattended. The desired Auto Connect date, day, or time values must be defined before Connect:Enterprise is brought online. When the defined date, day, or time is reached, Connect:Enterprise starts a connection with the remote sites listed in the ODF.

You can also initiate an Auto Connect session manually by using the:

- $$CONNECT console command
- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, using the $$CONNECT command, you can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program or the CICS or ISPF interface also enables you to override Auto Connect options set in the ODF. See *Activating and Overriding Auto Connect Sessions Manually* on page 156 for more information on initiating Auto Connects manually.

Auto Connect Processing

During an Auto Connect session, Connect:Enterprise can send batches to the remote site, receive batches from the remote site, or both send and receive in any order.

If the remote site rejects the Connect:Enterprise attempt to send batches, Connect:Enterprise instead attempts to receive batches from the remote site. After the batches are received and the disconnect interval expires, indicating that the remote site has finished sending, Connect:Enterprise again attempts to send batches. If the remote site again rejects the attempt to send batches, Connect:Enterprise again attempts to receive until the disconnect interval expires. This cycle repeats for three send/receive attempts; after that, the session is terminated. The Auto Connect report shows a transmit failure for each rejected attempt to send to the remote site. This could occur if the outbound batches were directed to an unavailable remote site printer.

Send Processing

The ways of identifying batches sent during an Auto Connect session are:

Method	Description
Standard Auto Connect	This method first sends batches that match the remote name, then sends batches that match the LISTNAME. These batches are then sent to all remote sites in the Auto Connect list before they are marked **T** (transmitted).
BEGINLIST parameter	Indicates the batch to be transmitted first. Specify the 1-8 character mailbox ID of the batch. This method could send a batch containing the BSC free-form SIGNON.
Auto Connect by IDLIST	Sends only those batches that match the mailbox IDs specified in the list.
$$CONNECT with ID specified	Sends all batches that match the mailbox ID in the $$CONNECT command. If BATCHID is also specified, all batches that match both ID and BATCHID are sent.

Batch Status Flags

Because you would not typically want to send batches multiple times for different Auto Connect sessions to the same remote site nor send batches that are no longer needed, Connect:Enterprise checks certain batch status flags before sending any batches. These batch status flags are the same as those displayed in $$DIRECTORY output, the ISPF interface, the CICS interface, or in offline utility LIST reports. The following criteria are used by Connect:Enterprise in determining whether a batch is transmitted during an Auto Connect session:

- The batch must be marked R (can be requested).

- The batch must not be marked T (already transmitted).
- The batch must not be marked D (delete).
- The batch must not be marked I (incomplete).
- The batch must not be transmit locked (added by the offline utilities with TRANSMITONCE=YES and then transmitted one time).

One exception to these rules enables you to send a batch that would not normally be sent for an Auto Connect session. If you want to force the retransmission of a batch marked T or I, you can enter its specific Mailbox ID and batch number in a $$CONNECT command from the console or through either the ISPF or CICS interface. See *Activating and Overriding Auto Connect Sessions Manually* on page 156.

When an Auto Connect session is activated but no batches meet the criteria for transmission, Connect:Enterprise sends the following message to the remote site:

```
*** NOTE *** TRANSMIT FAILED NO BATCHES FOR TRANSMISSION
DURING CONNECT:ENTERPRISE AUTO CONNECT.
```

The remote site still has the opportunity to send batches to Connect:Enterprise. For BSC sites, the remote site still has the opportunity to send batches if the MODE includes a RECV.

The NOBATCH=NC option in the Auto Connect list does not attempt a connection and does not send messages if no batches are available for transmission. The NOBATCH=NC feature is implemented for FTP Auto Connect sessions by the code in the LOGON_SCRIPT. See example member NOBATCH for sample REXX code.

Receive Processing

When Connect:Enterprise is receiving batches during an Auto Connect session, the remote site controls what constitutes a batch by the standard Connect:Enterprise $$ADD record. The mailbox ID specified on the $$ADD from the remote site does not have to match the remote name. However, if Connect:Enterprise batch security is used, the mailbox ID must be valid. Data received by Connect:Enterprise without a $$ADD record during an Auto Connect session uses the following default values:

```
ID=Remote Name from Auto Connect list
BATCHID=”AC BATCH WITHOUT $$ADD”
XMIT=N
```

Auto Connect receive processing is designed to receive data batches from remote sites with the host site initiating the connection. For this reason, the standard remote-initiated requests ($$REQUEST, $$DIRECTORY, and $$DELETE) are ignored during an Auto Connect receive.

Pending Processing

When Connect:Enterprise tries to start an Auto Connect session, it is possible that some remote sites in the Auto Connect list are in use by usual remote-initiated calls to the host site. If this is the case, Connect:Enterprise flags the required remote sites as pending Auto Connect sessions. As the remote sites become available, the Auto Connect list begins processing them. Keep in mind that excessively large remote-initiated processing can delay Auto Connect sessions in some cases.

A single remote site can never be shared by two separate Auto Connect sessions, so a pending state is not entered if a remote site is in use by another Auto Connect list. Any Auto Connect sessions that fail due to this condition display a console error message and are reported as failures in the Auto Connect report.

No pending condition is entered if you attempt to start more than one Auto Connect session for a listname which is already active. An attempted Auto Connect start for a listname that is in use fails and an appropriate error message is displayed unless Auto Connect queuing is in use for that listname.

FTP Script Processing

When an FTP Auto Connect session starts, Connect:Enterprise for z/OS identifies each remote server and executes the REXX script identified in the *REMOTES LOGON_SCRIPT definition. Typically, the LOGON_SCRIPT issues the logon commands (including a valid user ID and password) to connect to the remote server. You can put the logon commands in either the LOGON_SCRIPT or AC_SCRIPT, but putting them in the LOGON_SCRIPT simplifies logon and better controls password maintenance. (If you include the logon commands in the AC_SCRIPT, you must still have a LOGON_SCRIPT, although it would be blank in this case.)

If either the user ID or password are invalid, the connection fails. However, the script continues to run.

The LOGON_SCRIPT stops if any of the following occur:

- The script ends normally (through an exit or return) or the end of the script is reached.
- A REXX instruction syntax error occurs.
- The time between host command calls in a script exceeds the time specified in the SCRIPT_INTERVAL_TIME *OPTION parameter, and !TIMER OFF has not been issued since the last host command. (See Chapter 9, *FTP Auto Connect Scripts* for a description of the script commands.)

After the LOGON_SCRIPT successfully executes, the AC_SCRIPT is invoked and controls the remainder of the FTP session. If logon commands were not issued in the LOGON_SCRIPT, you must issue them in the AC_SCRIPT. AC_SCRIPT execution can end for the same reasons as the LOGON_SCRIPT.

The content and sequence of commands in either script is only restricted by the FTP protocol. Most FTP commands can be issued in any sequence after the session is established.

FTP Data Transmission

After logon, the Connect:Enterprise for z/OS client identifies itself and determines if the remote server is a Connect:Enterprise for z/OS server by issuing a SITE IDENT command. If the remote FTP server is also a Connect:Enterprise for z/OS server, Connect:Enterprise for z/OS logs the server information and executes proprietary protocol enhancement functions. This identification process is automatic, and runs immediately after logon, before other session commands are executed.

The format of the SITE IDENT command sent to the remote server is:

```
SITE IDENT PROD_ID=1 PROD_REL=x.x.00
```

where *x.x.00* is the release number of the Connect:Enterprise for z/OS version you are using. Some remote servers may reject the SITE IDENT command, but this will not end the session.

Queuing and Reactivating an Auto Connect Session

When an Auto Connect session cannot start, Connect:Enterprise queues the Auto Connect list and attempts to start it at a later time when its chance of success is greater. Auto Connect queuing activity is logged and reported with the REPORT utility.

Queuing is controlled by parameters set in the *OPTIONS record and the *CONNECT record. Setting ACQDEFAULT=Y in the *OPTIONS record activates queueing for all Auto Connect lists. You can change this default setting for an Auto Connect list by defining the ACQUEUE= parameter in the *CONNECT record for a specific Auto Connect list.

FTP Auto Connect lists are queued when no FTP thread is available for the session. The following table describes the conditions under which a queued FTP Auto Connect list is requeued or reactivated.

If Auto Connect list is queued because	Then it is requeued or reactivated when

ACQUEUE=Y
ACQDEFAULT=Y
	Overriding values are specified (for example, a $$CONNECT command is issued for the same listname but specifying a different mailbox ID)

An Auto Connect session is already running	Reactivated when the previous Auto Connect session ends
No FTP thread is available	Reactivated automatically when a thread becomes available
ACQUEUE=F (Force) for FTP Auto Connect	Queued unconditionally This option allows unlimited instances of an Auto Connect session to be queued without checking for any duplicate entries in the queue.

Activating and Overriding Auto Connect Sessions Manually

You can initiate an Auto Connect session manually by using the:

- $$CONNECT console command
- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, $$CONNECT can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program or the CICS or ISPF interface also enables you to override Auto Connect options set in the ODF. The manually activated command is useful if the data is not ready when the fully-automated Auto Connect session starts. The type of

Auto Connect session initiated depends on the operands used with the **$$CONNECT** command. The following example initiates a full Auto Connect session:

```
$$CONNECT L=LISTNAME
```

Auto Connect sessions can be manually activated at any time by entering the **$$CONNECT** command at an operator console, through the CICS interface or ISPF interface, or through a user-written CICS API program. You can type the following command on the system console or use the ISPF interface or CICS interface to initiate a partial Auto Connect session for a single Mailbox ID:

```
$$CONNECT L=xxxxxxxx ID=xxxxxxxx
```

Fully automated Auto Connect sessions process all remote sites in the *CONNECT list and send all batches with a Mailbox ID matching the remote name and list name, or the ID in the IDLIST parameter. However, you can use the $$CONNECT command to send a batch with a different mailbox ID to sites on an Auto Connect list, as illustrated in *Sending a Batch with a Different Mailbox ID to SNA Sites* on page 83.

Logging and Reporting Auto Connect Activity

Connect:Enterprise maintains a record of all batches sent and received during each Auto Connect session. As an Auto Connect session progresses, log records that describe the activity during the Auto Connect session are created in the VSAM log file. Auto Connect activity is reported by report utilities. The REPORT function in the offline utilities creates reports of activity during an Auto Connect session. The report utilities can run while Connect:Enterprise is online, and you can specify the type of data that is displayed on the report. The following table describes the contents and types of Auto Connect reports that are created.

Record	Description
Summary	Created for each Auto Connect session. The record contains information on the Auto Connect session, such as time and date started, time and date completed, number of successful batches transmitted and collected, and number of failed batches attempted. If an entire Auto Connect session fails, a failure reason code is recorded. If an entire Auto Connect session does not fail but one or more of the detail records have a failure code, the failure code from the first detail record is recorded in the summary record.
Detail	Created for each individual batch sent or received during the Auto Connect session. The record contains information for a single batch, such as time and date started, time and date completed, block count, remote name, Mailbox ID, user batch ID, and batch number. If any errors occur during the batch processing, a failure reason code is recorded.

Record	Description
Queued	Created .if an Auto Connect session is queued. The record contains information on the Auto Connect session, reason for queuing, and the time and date it was queued and reactivated. In addition, summary and detail records are written if no SNA session could be established. Use these records to determine if you must take corrective action before the automatic reactivation of the Auto Connect session

Auto Connect Console Messages

A console message is displayed whenever an Auto Connect session is initiated. See the *Connect:Enterprise for z/OS Messages and Codes Guide* for descriptions of Auto Connect messages.

If the Auto Connect session cannot start, a console message is issued. This message indicates if the Auto Connect session has been queued or has failed.

For SNA manual dial only, the console operator is prompted by VTAM to dial at the appropriate time. A console message is issued when the Auto Connect session actually gets under way.

When an Auto Connect session ends and all remote sites in the list have been accessed, a series of summary messages are written to the system console indicating the number of successful and failed transmissions and collections.

The REPORT function in the offline utilities enables you to analyze the Auto Connect session and determine what action is needed.

Configuring Records for FTP Auto Connect Sessions

This section describes configuring the *CONNECT record and parameters used to initiate FTP Auto Connect sessions. Chapter 7, *Configuring the *CALENDAR Record*, contains information on how to configure *CALENDAR records to define specific days, dates, or both on which to activate or deactivate time-initiated Auto Connect sessions. You can specify a calendar using the CALENDAR= parameter in the *CONNECT record.

Configuring the *CONNECT Record for FTP Auto Connect Lists

The *CONNECT record implements the Connect:Enterprise Auto Connect function. To use the Auto Connect function, specify a single *CONNECT record followed by one or more Auto Connect lists. The *CONNECT record consists of the following components: list name, list type, Auto Connect parameters, and remote site specification records. The *CONNECT parameters specify processing options for the Auto Connect session, such as time to initiate the session, number of concurrent sessions, and queueing. The remote site specification records used with the *CONNECT record specify the remote site, or sites, to contact and enable you to override certain site-specific parameters set in the *REMOTES record during the Auto Connect session. Each Auto Connect list is referred to by its LIST NAME. You can create an unlimited number of Auto Connect lists, and a single remote site can be included on multiple Auto Connect lists. The following example illustrates the structure of the *CONNECT record.

```
*CONNECT
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
    Auto Connect parameters
      Remote Site specification record
      Remote Site specification record
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
    Auto Connect parameters
      Remote Site specification record
      Remote Site specification record
```

Because Connect:Enterprise accesses the ODF every time the system is brought online, you can modify ODF values before you execute Connect:Enterprise. After Connect:Enterprise is online, you can activate an Auto Connect session by LISTNAME using the $$CONNECT console command, the ISPF interface, or the CICS interface at any time to temporarily override the ODF parameter values.

*CONNECT Record Format for FTP Auto Connect Lists

Before you configure an FTP Auto Connect list, review the rules in **CONNECT Record Rules* on page 159. The following example illustrates the *CONNECT record FTP Auto Connect parameters.

```
*CONNECT
  LISTNAME=XXXXXXX
   TYPE=FTP
   ACQUEUE=Y|N|F
   CALENDAR=xxxxxxxx
   SESSIONS=nnn|1
   TIME=hh:mm|[,hh:mm,...]
   Remote Site Specification Record
```

*CONNECT Record Rules

When you define the *CONNECT record, observe the following rules:

- *CONNECT must begin in column 1; any other text on that line is ignored.
- LISTNAME must be the first keyword; any other text on that line is ignored.
- The TYPE keyword must follow the LISTNAME keyword; any other text on the same line is ignored.
- Keywords can begin in any column and can include multiple values.
- Optional keywords can be in any order.
- To specify multiple values, separate the values by commas or blanks. If the multiple values do not fit in a single control record, repeat the keyword on a new control record.

*CONNECT Record Parameters for FTP Auto Connect Sessions

The following table lists *CONNECT parameters specific to FTP Auto Connect sessions. Required parameters are listed in bold first in the table; the remaining optional parameters are listed in alphabetical order. Defaults are underlined.

Parameter	Description
LISTNAME= XXXXXXXX	Required. Specifies the 1–8 character name of an Auto Connect list. If this value is defined as lowercase, Connect:Enterprise will force an uppercase value providing a consistent naming convention for duplicate LISTNAME verification. **Note:** The T-flag is not set on a batch with a matching Mailbox ID when that batch is flagged MULTXMIT in an FTP Client *REMOTES record and is transmitted during the FTP Auto Connect session.
TYPE=FTP	Required. Specifies that Connect:Enterprise acts as an FTP_CLIENT connecting to remote FTP servers in the Auto Connect list. This parameter must immediately follow LISTNAME=.

ACQUEUE=Y | N | F
ACQ=Y | N | F
	This parameter indicates whether the Auto Connect session should be queued and started later if the Auto Connect function cannot establish a session because another Auto Connect list is using the same name or no threads are available.
	If you do not specify a value, the default is determined by the value set for the ACQDEFAULT parameter in the *OPTIONS section of the ODF.
	YES = If an Auto Connect is started twice with the exact same parameters and same $$CONNECT overrides, the second Auto Connect is not queued.
	NO = Auto connect queueing does not occur.
	FORCE = The Auto Connect is queued unconditionally, that is, without a check for duplicate queue entries, if it cannot be activated immediately. You can also set on the $$CONNECT console command.
	Note: ACQDEFAULT=FORCE is not available in the *OPTIONS record.

CALENDAR=xxxxxxxx	Points to a calendar used for time-activated Auto Connect sessions. If specified, an entry must be defined in the *CALENDAR section of the ODF.
SESSIONS=nnn|<u>1</u>	Specifies the maximum number of concurrent sessions enabled during an Auto Connect session.
TIME=hh:mm[,hh:mm, ...]	Specifies one or more time-of-day values when Connect:Enterprise automatically activates a full Auto Connect session for the list. Specified as a 4-digit number separated by a colon, and a valid time using a 24-hour clock (for example, 08:00, 14:30). If this parameter is omitted, the only way to activate this list is to use the $$CONNECT console command or the CICS/ISPF interfaces. If the same time is specified for multiple Auto Connect sessions, they are spaced five seconds apart.

Add a Remote FTP Site to an FTP Auto Connect List

Following the Auto Connect session parameters, you must provide one or more remote site specification records. These records list each remote site accessed and additional options for each remote site.

FTP Remote Site Specification Record Format and Rules

The following example illustrates the format of the FTP remote site specification record. The remote site specification record parameters enable you to override the values set for certain parameters in the *REMOTES record. Default values for parameters are underlined.

```
REMOTE_NAME              AC_SCRIPT=name BCHSEP=NONE|OPT3|OPT4 ONEBATCH=YES|NO
                         &BEGINLIST=aaaaaaaa
                         &IDLIST=bbbbbbbb,cccccccc,...
                         &ENDLIST=zzzzzzzz
```

When you define FTP remote site specification records, observe the following rules:

- You must include at least one remote site specification record for an Auto Connect list.
- REMOTE_NAME is required and must be the first parameter specified in a remote site specification record.
- The REMOTE_NAME specified for an Auto Connect list must match a remote site name defined in the *REMOTES section of the ODF.
- Specify all optional parameters in any order on the same line as REMOTE_NAME; separate them by one or more spaces.
- The line containing REMOTE_NAME and optional parameters must precede &BEGINLIST, &IDLIST, and &ENDLIST.
- &BEGINLIST, &IDLIST, and &ENDLIST must be specified as the last parameters in an FTP remote site specification record in the following order: &BEGINLIST, &IDLIST, &ENDLIST.

FTP Remote Site Specification Record Parameters

The following table describes the FTP remote site specification record parameters. Required parameters are listed in bold first in the table. With the exception of the positional parameters, the remaining parameters are listed in alphabetical order. Acceptable abbreviations for parameters are enclosed in parentheses below the parameter in the following table.

Parameter	Description
REMOTE_NAME	Required. Specifies the name of the remote site. Must be the first operand on the remote site specification record. This parameter must match a name defined in the *REMOTES section of the ODF.
AC_SCRIPT=name	Specifies a member of a PDS that contains the AC_SCRIPT for this session. If a value is not specified, the default is determined by the setting of the FTP_AC_SCRIPT_DEFAULT parameter in the *OPTIONS section of the ODF. The specified member must be located in the PDS allocated to DD SYSEXEC when AC is run.

Parameter	Description
BCHSEP=<u>NONE</u> | OPT3 | OPT4	Specifies the method Connect:Enterprise uses to process batches sent to remote sites when multiple batches are sent in a single connection. All options of BCHSEP apply to Auto Connect session STOR or STOU commands transmitting multiple batches. This value can be changed within an AC_SCRIPT and in the remote definition. ◆ NONE—Concatenates all batches into a single file. During operation each batch is flagged transmitted after its last record is sent. A transmission failure after a batch is sent does not reset the T-flag for that batch and that batch remains unavailable for transmission until the operator intervenes. ◆ OPT3—Concatenates all batches into a single file. The individual batches are *not* flagged as transmitted until the entire transmission is successfully completed. A transmission failure after a batch is sent results in all batches remaining available for retransmission. The failure of STOR leaves all batches in that file available for retransmission even if some batches were successfully sent before the error. ◆ OPT4—Sends each batch as an individual file. Each batch is flagged T after transmission.
ONEBATCH=<u>NO</u> | YES (OB=Y | <u>N</u>)	Specifies whether only the first batch that meets the transmission criteria be sent. ◆ NO—Specifies that all batches matching transmission criteria are sent. ◆ YES—Specifies that only the first batch matching transmission criteria is sent.
&BEGINLIST= aaaaaaaa	Sets the value of the 1–8 character &BEGINLIST variable used in the AC_SCRIPT. Embedded blanks are not permitted. For this script variable to have the same function as the SNA/BSC &BEGINLIST keyword, program this action in your AC_SCRIPT. See the sample REXX List2, List3 and List4 in the sample data set.
&IDLIST=bbbbbbbb | [,cccccccc,...]	Sets the value of the 1–8 character &IDLIST variable used in the AC_SCRIPT. Specify one or more Mailbox IDs separated by commas or blanks. If all IDs do not fit on a single record, repeat the IDLIST keyword on subsequent records. For this script variable to have the same function as the SNA/BSC &IDLIST keyword, program this action in your AC_SCRIPT. See the sample REXX List2, List3 and List4 in the sample data set.
&ENDLIST=xxxxxxxx	Sets the value of the 1–8 character &ENDLIST variable used in the AC_SCRIPT. Embedded blanks are not permitted. For this script variable to have the same function as the SNA/BSC &ENDLIST keyword, program this action in your AC_SCRIPT. See the sample REXX List2, List3 and List4 in the sample data set.

The *CONNECT record has no DELAY, RETRY, DISCINTV, or NOBATCH parameters for FTP Auto Connect sessions because this function can be provided in the LOGON_SCRIPT. See the sample REXX members DELAY, RETRY, and NOBATCH in the sample data set.

Sample *CONNECT Record for FTP Auto Connect Lists

The following sample *CONNECT record is for FTP sites.

```
*CONNECT
  LISTNAME=LIST1
  TYPE=FTP
  ACQUEUE=Y
  SESSIONS=2
  TIME=08:00,14:00
  LASVEGAS  AC_SCRIPT=VEGAS BCHSEP=NONE ONEBATCH=Y
  NEWYORK  AC_SCRIPT=NYORK &BEGINLIST=YORK1
  LISTNAME=LIST2
  TYPE=FTP
  ACQUEUE=N
  TIME=09:00
  MEXICO &BEGINLIST=MEXBEG &IDLIST=MEX1,MEX2 &ENDLIST=MEXEND
```

In this example, the two FTP Auto Connect lists accomplish the following:

- LISTNAME=LIST1

  LIST1 is for FTP remote sites. The Auto Connect session is activated automatically at 8:00 a.m. and 2:00 p.m. each day. Two concurrent sessions are activated to allow Connect:Enterprise to communicate with both the Las Vegas site and the New York site simultaneously. If there are no FTP threads available, LIST1 is queued until a thread becomes available. The list connects to the remote sites in Las Vegas and New York. When Connect:Enterprise is communicating with the LASVEGAS remote site, the AC_SCRIPT VEGAS found in the PDS library allocated to DD SYSEXEC is used. When Connect:Enterprise communicates with the NEWYORK remote site, the AC_SCRIPT NYORK is used. The AC_SCRIPTs (VEGAS and NYORK in this case) determine whether the Auto Connect session sends or receives, and which batches are selected.

- LISTNAME=LIST2

  LIST2 is for FTP remote sites. The Auto Connect session is activated automatically at 9:00 a.m. each day. If there are no FTP threads available, LIST1 is not queued. The list connects to the remote site in Mexico. When Connect:Enterprise connects with the MEXICO remote site, the AC_SCRIPT identified in the *OPTIONS parameter FTP_AC_SCRIPT_DEFAULT= is run (since no AC_SCRIPT value is set on the remote specification record). The AC_SCRIPT determines whether the Auto Connect session sends or receives, and which batches are selected.

  For examples of using FTP scripts in ODF records, see *Sample FTP Options Definition Files* on page 163 and Chapter 9, *FTP Auto Connect Scripts*.

Sample FTP Options Definition Files

This section provides samples of ODFs for FTP connections.

Simple FTP AutoClient Connection

The following example shows a simple connection using FTP. All other *OPTIONS FTP parameters use the default values. No Auto Connect lists and no system security are used. Remote RMT001 can establish an FTP session with Connect:Enterprise as the server.

```
*OPTIONS
  FTP=YES
  APPLID=ENTPRS
  APPCPLSZ=0100
  VPF='CMBOX.VPF'
  APDSN='ENTPRS.ASSET.PROTECT.DATASET'
*REMOTES
  NAME=RMT001
        TYPE=FTP_CLIENT
```

Complex FTP Connection

The following example illustrates a more complex FTP connection. This FTP system uses FTP Auto Connect lists and SSL security is optional. Each day at 8:00 a.m., Auto Connect LIST1 runs. It connects to remote site MEXICO using logon REXX script GENLGN. Because the variables IPADDR and PORT are set, REXX GENLGN uses these variables to connect to the remote site. The LIST1 Auto Connect session also connects to remote site NEWYORK by running REXX GENLGN. The remote definitions for remote site NEWYORK specify different values for the PORT and IPADDR variables, which allows the session to use the same logon REXX while connecting to different remote sites. After the logon script for the MEXICO remote site runs, the AC script, also called MEXICO, runs. The REXX script MEXICO uses the variable DATATYPE and determines whether batches are collected or sent. After the logon script for the NEWYORK remote site runs, the AC script DFTAC, specified as the default Auto Connect script in the *OPTIONS section, runs. When LIST1 communicates with the LASVEGAS remote site, it uses default logon script DFTLGN and default AC script, DFTAC, because neither is specified in the remote specification record or the remote record. For more information on configuring FTP Auto Connect sessions, see *Configuring Records for FTP Auto Connect Sessions* on page 158.

```
*OPTIONS
  FTP=YES
  APPLID=ENTPRS
  APPCPLSZ=0100
  VPF='ENTPRS.VPF'
  APDSN='CMBOX.ASSET.PROTECT.DATASET'
  SCRIPT_INTERVAL_TIME=0060
  FTP_AC_SCRIPT_DEFAULT=DFTAC
  FTP_LOGON_SCRIPT_DEFAULT=DFTLGN
  SSL=YES
  SSL_KEY_DBASE_PW='SEC'
  SSL_KEY_DBASE='/U/KSTIC1/'cert.kdb’
  SSL_TIMEOUT=00300
  SSL_SERVER_CERT='SERVER1'
  SSL_CIPHER_SUITE=06010203
  SSL_DEFAULT_POLICY=OPTIONAL
  SSL_DEFAULT_CLIENT_AUTH_POLICY=OPTIONAL
*CONNECT
  LISTNAME=LIST1
   TYPE=FTP
  TIME=8:00
  MEXICO  AC_SCRIPT=MEXICO
  NEWYORK
   LASVEGAS
*REMOTES
  NAME=RMT001
   TYPE=FTP_CLIENT
  NAME=MEXICO
   TYPE=FTP_SERVER
         LOGON_SCRIPT=GENLGN
         &PORT=5565
         &IPADDR=10.34.154.7
  &DATATYPE=E
  SSL_POLICY=REQUIRED
  NAME=NEWYORK
      TYPE=FTP_SERVER
  LOGON_SCRIPT=GENLGN
  &PORT=5422
  &IPADDR=10.20.123.8
  NAME=LASVEGAS
   TYPE=FTP_SERVER
```

FTP Data Transfer Characteristics

FTP uses the data type, data structure, and transmission mode characteristics to determine how data is handled during an FTP transfer. The remote FTP client or server specifies these data characteristics; each FTP client or server can implement different characteristics. The following sections describe these characteristics.

Data Type

Data type specifies how the receiver interprets the data bits being transferred. The receiver can be any FTP client or FTP server. Connect:Enterprise FTP server assumes that the remote FTP client uses the ASCII character set. If the remote site does not use the ASCII character set, the remote FTP client must send the TYPE command to change the data type to EBCDIC or Image. Connect:Enterprise FTP supports the ASCII, EBCDIC, and Image data types, as described in the following table:

Data Type	Description
ASCII	Character data. The sender converts the data from its internal representation into standard 8-bit NVT-ASCII before sending it. Each line of data ends with a carriage-return-line-feed (<CRLF>). If Connect:Enterprise is receiving data, it translates the data from NVT-ASCII to EBCDIC and replaces the <CRLF> sequences with a new line (<NL>) character before adding the data to the current working mailbox. If Connect:Enterprise for z/OS is sending data, it translates the data from EBCDIC into NVT-ASCII and replaces the <NL> characters with the <CRLF> characters before transmitting the data to the remote FTP server. ASCII (TYPE A) is the default data type for Connect:Enterprise FTP.
EBCDIC	Character data. This is the most efficient data type for transfers between EBCDIC hosts. However, not all remote ASCII hosts accept data type EBCDIC. If Connect:Enterprise is receiving data, it translates the data before adding it to the current working mailbox. If Connect:Enterprise is sending data, Connect:Enterprise does not translate the data before transmitting it.
Image	Noncharacter data. Data is sent and received as contiguous bits packed into 8-bit bytes. Connect:Enterprise does not translate the data. Connect:Enterprise also assumes that the remote FTP client will not translate the data. You can transmit character data as the Image type, but it is not translated.

Data Structure

The FTP client specifies the structure of files transferred to and from Connect:Enterprise. Data structure is important when you transfer files between systems with different file storage methods. Some systems store files as file-oriented, whereas others store files as record-oriented. Connect:Enterprise FTP supports the file and record data structures, as described in the following table:

Data Structure	Description
FILE	The file has no internal structure and is considered a continuous sequence of bytes. File structure can be used with all transfer modes and data types. End-of-file (<EOF>) is indicated when the sender closes the data connection. File is the default Connect:Enterprise data structure.
RECORD	The data is sent as a set of sequential records. The record structure is only valid with text file transfers (data type ASCII or EBCDIC). The file must have explicit end-of-record (<EOR>) indicators for all records, including the final record.

Transfer Mode

Transfer mode indicates which transmission services Connect:Enterprise provides for FTP server. Connect:Enterprise supports the stream, block, and compressed transfer modes, as described in the following table:

Transfer Mode	Description
STREAM	Data is transferred as a stream of bytes. Any Connect:Enterprise data type or data structure can be used. Stream mode is the default transfer mode for the Connect:Enterprise server.
BLOCK	Data is transferred as a series of blocks. Any data type or file structure can use block mode transfer.
COMPRESSED	Data is transferred as a series of blocks in a compressed format. Only ASCII and EBCDIC data types can use compressed mode transfer. Only the file data structure can use compressed mode transfer. Compressed format transfers contain three kinds of information: uncompressed data, compressed data, and control information. ◆ Uncompressed data is transferred as a byte string, and is proceeded by a 1-byte count field. ◆ Compressed data can be either replication or filler. Replication compressed data is represented by a 1-byte count field preceding the replicated character. The count field is the number of times the single character occurs in the data. Filler compressed data is represented by a 1-byte count field. The count field is the number of filler characters that occur in the data. The filler byte is a blank character. ◆ Control information is sent as a 2-byte control sequence. The first byte is an escape (<ESC>) character. The second byte contains a descriptor code. Control information is optional.

The following table shows which transfer modes can be used for Connect:Enterprise FTP server data types and data structures:

Transfer Mode	Data Type			Data Structure	
	ASCII	EBCDIC	Image	File	Record
Stream	X	X	X	X	X
Block	X*	X	X*	X	X*
Compressed	X*	X	X*	X	X*

Note: The combinations marked with an asterisk (*) are not currently supported by the z/OS FTP server.

Connect:Enterprise FTP transfers data as 8-bit bytes. If the remote FTP server uses a different byte length, the remote FTP server must implement the proper conversion between the remote byte size and the Connect:Enterprise 8-bit byte transfer length.

Recommended Characteristics for Data Transfers

The following table shows the recommended data types, structures, and transfer mode for Connect:Enterprise data transfers:

Type of Data Transferred	Character Code Used by		Recommendation for Connect:Enterprise		
	Sending Host	Receiving Host	Data Type	Structure	Transfer Mode
Character	EBCDIC	EBCDIC	EBCDIC	RECORD	STREAM See Note 1.
		ASCII	ASCII	FILE	STREAM See Note 1.
	ASCII	EBCDIC	ASCII	FILE	STREAM See Note 1.
		ASCII	Image See Note 2.	FILE	STREAM
Binary	EBCDIC	EBCDIC	Image	RECORD	STREAM
		ASCII	Image	FILE	STREAM
	ASCII	EBCDIC	Image	FILE	STREAM
		ASCII	Image See Note 2.	FILE	STREAM

1 If the remote FTP server supports the standard FTP compression algorithm and the file is large enough to justify the extra CPU time involved with data compression and decompression, then use the compressed transfer mode.

2 Connect:Enterprise for z/OS operates on an EBCDIC character code host. This table entry implies that the data is sent to Connect:Enterprise for z/OS for intermediate storage and is eventually retrieved by another ASCII host.

All data transfers must complete with an end-of-file (<EOF>) that is specified in the data or implied when the sender closes the data connection.

When selecting a data type, structure, and transfer mode, consider the purpose of the file transfer. If the receiving host uses the transferred file, select an option that make the file usable on the receiving host. If the receiving host is an intermediate storage location for the transferred file, and the file is later retrieved again by the original host, select the Image as data type, file as data structure, and stream as transfer mode.

Troubleshooting

The follow table describes some common problems you may encounter when using FTP or SSL, and suggested corrective actions:

Problem	Action
The ODF is set up for SSL, but the following message is displayed: `CMB2200E-C:E RDV/TCP THRD=TCPCEMA1 REQ=SSLINIT RC=129 RS=ENOENT - No such file`	This is a UNIX error indicating that the pathname for the SSL key database cannot be found. Verify that the SSL key database name in the SSL_KEY_DBASE ODF *OPTIONS parameter is spelled correctly and is entered in the correct case.
The ODF is set up for SSL, but the following message is displayed: `CMB2200E-C:E RDV/TCP THRD=TCPCEMA1 REQ=SSLINIT RC=111 RS=EACCES - Permission is.`	This is a UNIX error indicating that the file permissions for the UNIX path specified in the ODF does not allow the Connect:Enterprise-assigned user ID read and write access. Use the CHMOD UNIX common to assign read and write permission to GROUP or OTHER.

The ODF is set up for FTP, but the following messages are displayed:
`ICH408I JOB(KSMBX1) STEP(KSMBX1) CL (PROCESS) OMVS SEGMENT NOT DEFINED`
`CMB2181E-Connect:Enterprise/TCP TCP/IP REQUEST BPX1GCW FAILED WITH RC=009C, RS=00F9.`
`CMB2200E-Connect:Enterprise/TCP REQ=BPX1GCW RC=00156 RS=Unix Proc Init Error`
	The user ID assigned to the Connect:Enterprise system does not have permission to use UNIX facilities. Connect:Enterprise requires UNIX facilities for FTP operation. Contact your security administrator.

The ODF is set up for SSL, but the following message is displayed:
`CMB2200E-Connect:Enterprise/TCP THRD=FTPS0001 REQ=SSLSOINI RC=-41 RS=GSK_SOC_BAD_V3_CIPHER`
`If you turn on tracing, the following is displayed:`
`2000.126 12:39:23 8b00b0 SSL cipher-suite code: 0A is unavailable.`
	The messages indicate that some of the cipher suites listed in the SSL_CIPHER_SUITE ODF *OPTIONS parameter are invalid or not supported by IBM software. Verify that the cipher suites are correct, or contact IBM to order and install the correct level of system SSL support.

The ODF is set up for SSL, but the following message is displayed: `CMB2200E-Connect:Enterprise/TCP THRD=FTPS0001 REQ=SSLSOINI RC=-1 RS=GSK_SOC_ERROR_NO_CIPHERS`	This message indicates that a common cipher suite could not be negotiated between Connect:Enterprise and the remote client. Change either the SSL_CIPHER_SUITE ODF *OPTIONS parameter or the remote client to use a common cipher suite.
The ODF is set up for SSL, but the following messages are displayed: ◆ `CMB2200E-Connect:Enterprise/TCP THRD=FTPS0001 REQ=SSLRECV RC=1121 RS=Connection reset by peer` ◆ `CMB2200E-Connect:Enterprise/TCP THRD=FTPS0001 REQ=SSLSOINI RC=-10 RS=GSK_ERROR_IO` ◆ `CMB2181E-Connect:Enterprise/TCP REQUEST TCPRECV FAILED WITH RC=0461, RS=7242`	These messages indicate that the remote client could not validate your certificate using the currently configured trusted root file. This may occur when self-signed certificates are used. If using self-signed certificates, place a copy of the certificate on the workstation and configure the client to use it for the trusted root file.

Chapter 6

Configuring ODF Records for BSC Connections

This chapter describes configuring the *OPTIONS, *SECURITY, *SIGNON, *IDVER, and *CONNECT records for BSC connections.

Defining *OPTIONS Parameters for BSC Connections

The BSC parameters set in the *OPTIONS record enable BSC communications and set global, default values for both host-initiated and remote-initiated BSC connections. These values define the attributes of BSC sessions unless they are overridden by equivalent parameters set from the command line or from one of the user interfaces.

Before you begin configuring the parameters for remote BSC connections, review the *OPTIONS record format and the rules for defining *OPTIONS parameters in Chapter 3, *Configuring *OPTIONS Record for System Resources*. The following table lists the *OPTIONS record parameter definitions specific to BSC connections. Required parameters are listed in bold first in the table; the remaining parameters are listed alphabetically.

Parameter	Description
BSC_DEFAULT_ $$DIR_FORMAT = BID24 \| BID64	Specifies how Connect:Enterprise formats the reply to a $$DIR command during a Bisync session. The default is set by the DEFAULT_MODE=BID24|BID64 parameter. If BID24 is specified, BID24 is the default for this parameter; if BID64 is specified, BID64 is used for this parameter. ◆ BID24—The directory display is generated, using the first 24 characters of the User Batch ID. ◆ BID64—The directory display is generated, using the full 64 characters of the User Batch ID. This value can be overridden on a per command basis, by specifying the FORMAT=BID24 | BID64 parameter in the $$DIR command record.
BTAM=YES	Required to activate the BTAM telecommunications method.
RETAIN	Used with BSC remote sites that use the $$ADD command and do not specify all of the required parameters for that command. The value for the unspecified parameters is obtained from the previous $$ADD command issued during that session.
RMDC=YES	Invokes the Connect:Enterprise capability to Receive Multiple Data Collections on switched lines. When Connect:Enterprise is receiving data during the data collection process, it separates the data into multiple batches if the proper BSC line control is used. This is only valid for BSC. Permits remote sites to send multiple batches to Connect:Enterprise using file separation of ETX on the last transmission block of one file followed by EOT and then ENQ to start the next file. Connect:Enterprise continues to collect batches until two EOTs are received sequentially, a DLEEOT (disconnect) is received, or until four read timeouts occur. This parameter is in effect for batch collections occurring during the RECEIVE phase of a remote connect session or an Auto Connect session with MODE=SENDRECV, RECVSEND, or RECVONLY. It is also in effect for batches collected after a $TURNLINE$=RMDC. It is not in effect for a collection sequence after a $TURNLINE$, which ends upon receiving one EOT. Without RMDC=YES, multiple batches are collected within a single RECEIVE if they are separated by ETXs or $$ADD control records beginning the first block of each batch. RMDC=YES is not required for JES communications. Since JES can always send Connect:Enterprise multiple batches separated by EOTs, Connect:Enterprise always assumes RMDC=YES when communicating with JES.
SCINCOR=YES | <u>NO</u>	If SECURITY=BATCH, this parameter must specify whether the mailbox IDs are maintained in memory or read from the ODF for each mailbox ID validation. ◆ YES—The mailbox IDs are in memory. Each ID requires only 8 bytes of storage. This is the recommended value. ◆ NO—The mailbox IDs are read from the ODF.

Parameter	Description
SECURITY=BATCH	When Connect:Enterprise connects with BSC sites, it can specify only one SECURITY command: SECURITY=BATCH ◆ BATCH—Ensures that all transactions transmitted from remote terminals are processed only if a valid mailbox ID is supplied by the remote site as part of the transmission. Valid IDs are listed in the *SECURITY record. See *Defining *SECURITY Record Parameters for BSC Connections* on page 173.
UA=xxxxxxxx	Required if BTAM=YES. Specifies the load module name of your custom User Assembly, which defines your BTAM network to Connect:Enterprise. Place this module in your JOBLIB or your STEPLIB for Online Connect:Enterprise.
WACKMAX=20|nnn	Sets the maximum limit of BSC WACKs (X'106B') received from a communicating partner while Connect:Enterprise waits for the transmission to continue. Connect:Enterprise responds with the appropriate ENQ (X'2D') until the limit has been exhausted. The default of 20 is not adequate for some connections when a remote site responds with many WACKs before continuing a session. The maximum value is 255 sequential WACKs. The default is 20.

Defining *SECURITY Record Parameters for BSC Connections

The *SECURITY record is not required for remote BSC connections unless you define SECURITY=BATCH in the *OPTIONS record, which enables you to implement batch security for remote connections from remote BSC sites without implementing the Connect:Enterprise for z/OS security interface. If you are implementing batch security, you must include the 80-byte *SECURITY record immediately following the *OPTIONS record. The *SECURITY record lists the valid mailbox IDs for your system and restricts remote users to the mailbox ID assigned to their site.

If you do not implement batch security for BSC connections, you can go to *Configuring the *SIGNON Record for Remote BSC Sites* on page 174.

*SECURITY Record Format and Parameters

The *SECURITY record is followed immediately by records containing the valid mailbox IDs for your system. The mailbox ID is 1–8 characters with no embedded blanks. The following example is a valid ID specification:

```
*SECURITY
       ID=xxxxxxxx
```

The following example shows a record with multiple mailbox IDs separated by commas.

```
*SECURITY
       ID=xxxxxxxx,ID=xxxxxxxx,ID=xxxxxxxx
```

*SECURITY Record Rules

When you configure the *SECURITY record, observe the following rules:

- When SECURITY=BATCH in the *OPTIONS record, the *SECURITY record is required and must follow the *OPTIONS record.
- *SECURITY must begin in column 1; any other text on the same line is ignored.
- Specify multiple mailbox IDs in a single *SECURITY record.
- Separate mailbox IDs by either commas or blanks.

Configuring the *SIGNON Record for Remote BSC Sites

The *SIGNON record is optional and used to identify the valid SIGNON formats when the remote site sends a signon record to the host when a transmission connection is established. Remote terminals that communicate with Connect:Enterprise are often designed to communicate with special Remote Job Entry (RJE) Systems, such as JES, at the host. Some systems require a SIGNON record from the remote site when the transmission connection is established.

*SIGNON Record Rules

When you define the *SIGNON record, observer the following rules:

- Include a *SIGNON record in the ODF if the remote site uses SIGNON records.
- *SIGNON must begin in column 1; any other the text on the same line is ignored.
- The *SIGNON control record must be followed by one or more records defining the exact SIGNON data format used by remote sites.
- Connect:Enterprise supports two methods for specifying valid signons:
  - Fixed-form BSC SIGNON
  - Free-form BSC SIGNON

Fixed-Form BSC SIGNON

Connect:Enterprise does not require the SIGNON record. However, Connect:Enterprise accepts the SIGNON and responds to it if necessary because some remote terminals always send a SIGNON. Connect:Enterprise permits any format that includes six characters for the SIGNON records from the remote site, and does not validate or use the data in any way. However, Connect:Enterprise must determine that a SIGNON record is part of the connection Process and not part of the data being collected. If a remote site sends SIGNON records to a host, define the valid SIGNON record formats using the *SIGNON control record in the ODF.

Fixed-Form *SIGNON Example

If remote sites use two possible SIGNON records (/*SIGNON or $SIGNON), use the following records in the ODF:

```
*SIGNON
/*SIGNON
$SIGNON
```

The *SIGNON record is a Connect:Enterprise ODF control record and not a valid SIGNON data record. If your remote terminals use *SIGNON as a SIGNON data record, include the second *SIGNON after the *SIGNON control record.

Free-Form BSC SIGNON

An optional free-form BSC SIGNON can be used by Connect:Enterprise for remote identification and security checking. Free-form SIGNON definitions differ from fixed-form SIGNON definitions in that the remote name and passwords are not hard coded and the positions of each are identified so the values can be extracted and used in later processing. The use of masked positions provides increased security for the remote name and password. Additionally, a remote name supplied in the SIGNON is used as the default Mailbox ID for data collections which do not specify a mailbox ID. This does not interfere with the standard use of SIGNON. The name and password can be 1–8 characters.

Connect:Enterprise identifies the remote name and password information when using free-form BSC SIGNON, passes it to the Security exit one, and then it passes it to the security interface. This user-supplied security exit examines the remote name and passwords (or, in the case of a standard BSC SIGNON, the lack of a name and password) and determines if processing can continue. The SIGNON record is only passed through to the optional security processing and is *not* maintained by Connect:Enterprise.

If the SIGNON record is present and the remote name and password are identified by Connect:Enterprise, the remote connect log records contain the remote name for better identification of the remote site.

Using the Free-Form BSC SIGNON for Remote-Initiated Connections

The use of the BSC SIGNON feature for remote-initiated connections is optional and is only invoked when the *SIGNON section of the ODF contains records with the special mask characters. One or more SIGNON model records can be supplied, with the standard SIGNON data and the mask characters in different positions as needed.

The following table lists the special characters used for the mask:

Special Characters	Position
########	Remote name
%%%%%%%%	Password

Special Characters	Position
++++++++	New password

The mask character string is eight characters long. The data that is sent in the mask positions is 1–8 characters, left-justified and blank-filled. Only the remote name mask is required. If a password mask is not specified, or if no data is sent in a mask area for the password, the user exits are called with blanks in the password parameters.

All other nonmasked text must match before the SIGNON record is matched and the masked fields are extracted.

The following are examples of SIGNON records and the associated SIGNON model record supplied in the *SIGNON section of the ODF.

The following is an example of a standard JES2 format SIGNON:

```
/*SIGNON       remotnam       newpassw                      password
```

The following is the *SIGNON section model record:

```
/*SIGNON       ########       ++++++++                      %%%%%%%%
```

The following is an example of a company’s standard SIGNON:

```
$$SIGNON REMOTE=remotnam PASSWORD=password
```

The following is the *SIGNON section model record:

```
$$SIGNON REMOTE=######## PASSWORD=%%%%%%%%
```

Configuring the *IDVER Record for BSC Connections

*IDVER records must be supplied if any lines defined to Connect:Enterprise specified the BTAM ID verification option (M$LINEX ...,IDVER=).

*IDVER Record Rules and Format

When you configure the *IDVER record, observe the following rules:

- *IDVER must begin in column one and the remainder of the record is ignored.

- If the host ID is to be exchanged during BTAM ID verification (M$LINEX specified IDVER=HOST or IDVER=BOTH), the *IDVER record is followed immediately by a record that contains only the host ID used. The 1–15 character host ID contains no embedded blanks. A valid host ID is specified as shown:

```
HID=xxx...xxx
```

- If remote IDs are exchanged during BTAM ID verification (M$LINEX specified IDVER=REMOTE or IDVER=BOTH), the *IDVER section must contain a series of records that specify all possible valid remote IDs. 1–18 character remote IDs contain no embedded blanks. BTAM restricts the number of possible remote IDs to 192. If your system contains more than 192 remote sites, some must use duplicate remote IDs. For example, valid remote IDs can be specified on one or more records, separated by blanks or commas. The following is an example of IDs separated by blanks:

```
RID=xxx...xxx RID=xxx...xxx RID=xxx...xxx
```

*IDVER Example

In this example, the host ID is ENTPRS. The remote IDs are city names followed by a three-digit number that uniquely identifies the remote site. The ODF *IDVER section could contain the following:

```
*IDVER
    HID=ENTPRS
    RID=DALLAS001, RID=DALLAS002, RID=DALLAS003
    RID=HOUSTON01, RID=HOUSTON02,
    RID=SANANTOIO999
    RID=ELPAS0007
```

Verifying the BTAM ID

Use the following procedure to verify the BTAM ID:

1. Specify IDVER=BOTH, IDVER=HOST, or IDVER=REMOTE on the user assembly M$LINEX macros for lines that use BTAM ID verification.
2. Create the *IDVER section of the ODF. Use the HID parameter to specify a single host ID, and the RID parameters to specify one or more remote IDs.

   For BTAM ID verification during an Auto Connect session, include the RID parameter on the remote specification record in the *CONNECT section.
3. Supply the remote sites authorized to use Connect:Enterprise with their unique remote ID (defined in RID) and the host ID (defined in HID).
4. In rare cases, you must provide different BTAM ID verification capabilities on different lines. This enables remote sites with different BTAM ID verification implementations to access Connect:Enterprise. However, do not compromise security in your network.

If mixed mode BTAM ID verification is used, provide the proper host telephone numbers to your remote sites, depending on the type of ID verification done by the remote site.

A Connect:Enterprise Auto Connect session with BTAM ID verification requires all remote sites in the same Auto Connect list to use the same method of BTAM ID verification. You must ensure that a compatible line is assigned to the Auto Connect session by specifying the line ID in the LINES parameter of the Auto Connect list.

5. You can supply a Security Exit Two load module which Connect:Enterprise executes if a BTAM ID verification error occurs. You must code and test this program to keep track of the errors. However, do not attempt to ignore or override a BTAM ID verification error because BTAM, not Connect:Enterprise, verifies the ID and forces a line disconnect if errors occur.

   For BTAM ID verification errors, the security violation code is set to C'2' and the X2$BCHID field contains the invalid ID received from the remote site.

About Auto Connect Sessions

An Auto Connect initiates the connection between the host site and the remote site. An AutoConnect session is either fully automated or manually initiated. Both data transmission and data collection can be performed during an Auto Connect session. For BSC sites, an Auto Connect session can use switched auto dial, switched manual dial, and nonswitched lines. A fully automated Auto Connect session cannot be used with manual dial lines because operator intervention is required for dialing. An Auto Connect Manager (ACM) is responsible for the Auto Connect session, and ACM tasks can be replicated to allow for processing multiple concurrent requests.

Fully automated Auto Connect sessions are activated each day when the system clock reaches the time of day specified in an Auto Connect list. If Connect:Enterprise remains up for multiple days, the Auto Connect session is activated every day when the system clock reaches the specified time. You can also define *CALENDAR records, and refer to them in the *CONNECT record, to specify dates and days of the week on which to activate or deactivate Auto Connect processing. See Chapter 7, *Configuring *CALENDAR Records* for details.

A fully automated Auto Connect session is initiated by a date, day, or time specified in:

- *CONNECT record in the ODF
- User-written CICS API program

After it is set up, a fully automated Auto Connect session does not need operator intervention at the host site or the remote site, if the hardware at both sites can operate unattended. The desired Auto Connect date, day, or time values must be defined before Connect:Enterprise is brought online. When the defined date, day, or time is reached, Connect:Enterprise starts a connection with the remote sites listed in the ODF.

You can also initiate an Auto Connect session manually by using the:

- $$CONNECT console command

- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, using the $$CONNECT command, you can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program, the CICS interface, or the ISPF interface also enable you to override Auto Connect options set in the ODF. See *Activating and Overriding Auto Connect Sessions Manually* on page 181 for more information on initiating Auto Connects manually.

Auto Connect Processing

During an Auto Connect session, Connect:Enterprise can send batches to the remote site, receive batches from the remote site, or both send and receive in any order.

At SNA sites, Connect:Enterprise always attempts to send batches first. Normally, all batches available for transmission to the remote site are sent immediately. Connect:Enterprise then tries to receive batches from the remote site. Connect:Enterprise continues to receive batches until the disconnect interval expires, indicating that the remote is finished sending, or until the remote site ends the session.

If the remote site rejects the Connect:Enterprise attempt to send batches, Connect:Enterprise instead attempts to receive batches from the remote site. After the batches are received and the disconnect interval expires, indicating that the remote site has finished sending, Connect:Enterprise again attempts to send batches. If the remote site again rejects the attempt to send batches, Connect:Enterprise again attempts to receive until the disconnect interval expires. This cycle repeats for three send/receive attempts; after that, the session is terminated. The Auto Connect report shows a transmit failure for each rejected attempt to send to the remote site. This could occur if the outbound batches were directed to an unavailable remote site printer.

Send Processing

The ways of identifying batches sent during an Auto Connect session are:

Method	Description
Standard Auto Connect	This method first sends batches that match the remote name, then sends batches that match the LISTNAME. These batches are then sent to all remote sites in the Auto Connect list before they are marked **T** (transmitted).
BEGINLIST parameter	Indicates the batch to be transmitted first. Specify the 1-8 character mailbox ID of the batch. This method could send a batch containing the BSC free-form SIGNON.
Auto Connect by IDLIST	Sends only those batches that match the mailbox IDs specified in the list.
$$CONNECT with ID specified	Sends all batches that match the mailbox ID in the $$CONNECT command. If BATCHID is also specified, all batches that match both ID and BATCHID are sent.

Batch Status Flags

Because you would not typically want to send batches multiple times for different Auto Connect sessions to the same remote site nor send batches that are no longer needed, Connect:Enterprise checks certain batch status flags before sending any batches. These batch status flags are the same as those displayed in $$DIRECTORY output, the ISPF interface, the CICS interface, or in offline utility LIST reports. The following criteria are used by Connect:Enterprise in determining whether a batch is transmitted during an Auto Connect session:

- The batch must be marked R (can be requested).
- The batch must not be marked T (already transmitted).
- The batch must not be marked D (delete).
- The batch must not be marked I (incomplete).
- The batch must not be transmit locked (added by the offline utilities with TRANSMITONCE=YES and then transmitted one time).

One exception to these rules enables you to send a batch that would not normally be sent for an Auto Connect session. If you want to force the retransmission of a batch marked T or I, you can enter its specific mailbox ID and batch number in a $$CONNECT command from the console or through either the ISPF or CICS interface. See *Activating and Overriding Auto Connect Sessions Manually* on page 181.

When an Auto Connect session is activated but no batches meet the criteria for transmission, Connect:Enterprise sends the following message to the remote site:

```
*** NOTE *** TRANSMIT FAILED NO BATCHES FOR TRANSMISSION
DURING CONNECT:ENTERPRISE AUTO CONNECT.
```

The remote site still has the opportunity to send batches to Connect:Enterprise. For BSC sites, the remote site still has the opportunity to send batches if the MODE includes a RECV.

The NOBATCH=NC option in the Auto Connect list does not attempt a connection and does not send messages if no batches are available for transmission. The NOBATCH=NC feature is implemented for FTP Auto Connect sessions by the code in the LOGON_SCRIPT. See example member NOBATCH for sample REXX code.

Receive Processing

When Connect:Enterprise is receiving batches during an Auto Connect session, the remote site controls what constitutes a batch by the standard Connect:Enterprise $$ADD record. The mailbox ID specified on the $$ADD from the remote site does not have to match the remote name. However, if Connect:Enterprise batch security is used, the mailbox ID must be valid. Data received by Connect:Enterprise without a $$ADD record during an Auto Connect session uses the following default values:

```
ID=Remote Name from Auto Connect list
BATCHID=”AC BATCH WITHOUT $$ADD”
XMIT=N
```

Auto Connect receive processing is designed to receive data batches from remote sites with the host site initiating the connection. For this reason, the standard remote initiated requests ($$REQUEST, $$DIRECTORY, and $$DELETE) are ignored during an Auto Connect receive.

At SNA sites, a $$LOGOFF command can be sent to the host if the remote site wants to end the session at any time.

Pending Processing

When Connect:Enterprise tries to start an Auto Connect session, it is possible that some remote sites in the Auto Connect list are in use by usual remote-initiated calls to the host site. If this is the case, Connect:Enterprise flags the required remote sites as pending Auto Connect sessions. As the remote sites become available, the Auto Connect list begins processing them. Keep in mind that excessively large remote-initiated processing can delay Auto Connect sessions in some cases.

A single remote site can never be shared by two separate Auto Connect sessions, so a pending state is not entered if a remote site is in use by another Auto Connect list. Any Auto Connect sessions that fail due to this condition display a console error message and are reported as failures in the Auto Connect report.

No pending condition is entered if you attempt to start more than one Auto Connect session for a listname which is already active. An attempted Auto Connect start for a listname that is in use fails and an appropriate error message is displayed unless Auto Connect queuing is in use for that listname.

Queuing and Reactivating an Auto Connect Session

When an Auto Connect session cannot start, Connect:Enterprise queues the Auto Connect list and attempts to start it at a later time when its chance of success is greater. Auto Connect queuing activity is logged and reported with the REPORT utility.

Queuing is controlled by parameters set in the *OPTIONS record and the *CONNECT record. Setting ACQDEFAULT=Y in the *OPTIONS record activates queueing for all Auto Connect lists. You can change this default setting for an Auto Connect list by defining the ACQUEUE= parameter in the *CONNECT record for a specific Auto Connect list.

BSC Auto Connect lists are queued and requeued (or reactivated) in the following situations:

- A BSC Auto Connect list is already running.
- No BSC lines are available for an Auto Connect session.
- When an Auto Connect list is queued because ACQUEUE=Y or ACQDEFAULT=Y is set, and overriding values are specified (for example, a $$CON command is issued for the same list but specifies a different mailbox ID), then it is requeued or reactivated.

Activating and Overriding Auto Connect Sessions Manually

You can initiate an Auto Connect session manually by using the:

- $$CONNECT console command
- CICS interface
- ISPF interface

The $$CONNECT command provides Auto Connect options and overrides. For example, $$CONNECT can initiate a full Auto Connect session or transmit specific batches to the remote sites in the Auto Connect list. A user-written CICS API program or the CICS or ISPF interface also enables you to override Auto Connect options set in the ODF. The manually activated command is useful if the data is not ready when the fully-automated Auto Connect session starts. The type of Auto Connect session initiated depends on the operands used with the **$$CONNECT** command. The following example initiates a full Auto Connect session:

```
$$CONNECT L=LISTNAME
```

Auto Connect sessions can be manually activated at any time by entering the **$$CONNECT** command at an operator console, through the CICS interface or ISPF interface, or through a user-written CICS API program. You can type the following command on the system console or use the ISPF interface or CICS interface to initiate a partial Auto Connect session for a single mailbox ID:

```
$$CONNECT L=xxxxxxxx ID=xxxxxxxx
```

Fully automated Auto Connect sessions process all remote sites in the *CONNECT list and send all batches with a mailbox ID matching the remote name and list name, or the ID in the IDLIST parameter. However, you can use the $$CONNECT command to send a batch with a different mailbox ID to sites on an Auto Connect list, as illustrated in *Sending a Batch with a Different Mailbox ID to BSC Sites* on page 182.

Sending a Batch with a Different Mailbox ID to BSC Sites

This example assumes that the remote site can respond to the MODE override and accept a host-initiated send. If the remote site must always send first but is capable of sending an indication that no data exists for transmission to the host, then MODE=RECVSEND can be used in the $$CONNECT, the ISPF interface, or the CICS interface instead of SENDONLY. Connect:Enterprise can never force a remote site to properly respond to a defined MODE, so a MODE must be used that is compatible with the capabilities of the remote site.

```
$$CONNECT L=ECOAST ID=ALERT MODE=SENDONLY
$$CONNECT L=WCOAST ID=ALERT MODE=SENDONLY
```

See the *Console Commands* chapter in the *Connect:Enterprise for z/OS for z/OS User's Guide* for a description of the $$CONNECT console command. See the *Connect:Enterprise for z/OS CICS User's Guide* to use the Auto Connect feature with CICS. See the *Connect:Enterprise for z/OS ISPF User's Guide* to use the Auto Connect feature with ISPF.

Logging and Reporting Auto Connect Activity

Connect:Enterprise maintains a record of all batches sent and received during each Auto Connect session. As an Auto Connect session progresses, log records that describe the activity during the Auto Connect session are created in the VSAM log file. Auto Connect activity is reported by report utilities. The REPORT function in the offline utilities creates reports of activity during an Auto Connect session. The report utilities can run while Connect:Enterprise is online, and you can specify the type of data that is displayed on the report. The following table describes the contents and types of Auto Connect reports that are created.

Record	Description
Summary	Created for each Auto Connect session. The record contains information on the Auto Connect session, such as time and date started, time and date completed, number of successful batches transmitted and collected, and number of failed batches attempted. If an entire Auto Connect session fails, a failure reason code is recorded. If an entire Auto Connect session does not fail but one or more of the detail records have a failure code, the failure code from the first detail record is recorded in the summary record.
Detail	Created for each individual batch sent or received during the Auto Connect session. The record contains information for a single batch, such as time and date started, time and date completed, block count, remote name, mailbox ID, user batch ID, and batch number. If any errors occur during the batch processing, a failure reason code is recorded.
Queued	Created if an Auto Connect session is queued. The record contains information on the Auto Connect session, reason for queuing, and the time and date it was queued and reactivated. In addition, summary and detail records are written if no SNA session could be established. Use these records to determine if you must take corrective action before the automatic reactivation of the Auto Connect session

Auto Connect Console Messages

A console message is displayed whenever an Auto Connect session is initiated. See the *Connect:Enterprise for z/OS Messages and Codes Guide* for descriptions of Auto Connect messages.

If the Auto Connect session cannot start, a console message is issued. This message indicates if the Auto Connect session has been queued or has failed.

For SNA manual dial only, the console operator is prompted by VTAM to dial at the appropriate time. A console message is issued when the Auto Connect session actually gets under way.

When an Auto Connect session ends and all remote sites in the list have been accessed, a series of summary messages are written to the system console indicating the number of successful and failed transmissions and collections.

The REPORT function in the offline utilities enables you to analyze the Auto Connect session and determine what action is needed.

Configuring the *CONNECT Record for BSC Auto Connect Lists

The *CONNECT record implements the Connect:Enterprise Auto Connect function. The *CONNECT record consists of the following components: list name, list type, Auto Connect parameters, and remote site specification records. The *CONNECT parameters specify the name and type of the list, and processing options for the Auto Connect session, such as time to initiate the session, number of concurrent sessions, and queueing. The remote site specification records used with the *CONNECT record specify the remote site, or sites, to contact and site-specific parameters for the Auto Connect session.

To use the Auto Connect function, specify a single *CONNECT record followed by one or more Auto Connect lists. Each Auto Connect list is referred to by its LISTNAME. You can create an unlimited number of Auto Connect lists, and a single remote site can be included on multiple Auto Connect lists. The following example illustrates the structure of the *CONNECT record.

```
*CONNECT
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
     Auto Connect parameters
       Remote Site specification record
       Remote Site specification record
  LISTNAME=XXXXXXXX
  TYPE=XXXXXX
     Auto Connect parameters
       Remote Site specification record
       Remote Site specification record
```

Because Connect:Enterprise accesses the ODF every time the system is brought online, you can modify ODF values before you execute Connect:Enterprise. After Connect:Enterprise is online, you can activate an Auto Connect session by LISTNAME using the $$CONNECT console command, the ISPF interface, or the CICS interface at any time to temporarily override the ODF parameter values.

*CONNECT Record Format for BSC Auto Connect Lists

Before you configure a BSC Auto Connect list, review the rules in *Sample BSC Options Definition Files* on page 200. The following example illustrates the *CONNECT record format for BSC Auto Connects.

```
LISTNAME=XXXXXXXX
TYPE=BSCAD | BCSMD | BSCNS
ACQUEUE=Y | N
CALENDAR=xxxxxxxx
DELAY=0 |nnnn
DISCINTV=NO |nnnn | 0
JES=NO | YES
LINES=xxxxxxxx[,xxxxxxxx]
NOBATCH=C | NC
POWER=NO | YES
RETRY=0 | nn
SIGNOFF=NO | YES
TIME=hh:mm[,hh:mm,...]
Remote Site Specification Record
```

*CONNECT Record Rules

When you define the *CONNECT record, observe the following rules:

- *CONNECT must begin in column 1; any other text on that line is ignored.
- LISTNAME must be the first keyword; any other text on that line is ignored.
- The TYPE keyword must follow the LISTNAME keyword; any other text on the same line is ignored.
- Keywords can begin in any column and can include multiple values.
- Optional keywords can be in any order.
- To specify multiple values, separate the values by commas or blanks. If the multiple values do not fit in a single control record, repeat the keyword on a new control record.

*CONNECT Record Parameters for BSC Auto Connects

The following table list the *CONNECT parameter definitions for BSC Auto Connects. Required parameters are listed in bold first in the table; the remaining optional parameters are listed in alphabetical order. Defaults are underlined.

Parameter	Description
LISTNAME=XXXXXXXX	Required. Specifies the 1–8 character name of an Auto Connect list. If this value is defined as lowercase, Connect:Enterprise will force an uppercase value providing a consistent naming convention for duplicate LISTNAME verification.

Parameter	Description
TYPE=BSCAD | BSCMD | BSCNS	Required. Specifies the type of session for the Auto Connect session. This parameter must immediately follow LISTNAME=. BSC connections require one of the following options: ◆ BSCAD—Uses BSC auto-dial line. ◆ BSCMD—Uses BSC manual dial line. Cannot use a fully automated Auto Connect session. ◆ BSCNS—Uses BSC nonswitched lines.
ACQUEUE=Y | N	Indicates whether the Auto Connect session should be queued and started later if the Auto Connect function cannot establish a session with at least one remote site. If you specify N, the Auto Connect function fails if resources are not available at the time it is initiated. If you do not specify a value, the default is determined by the setting of the ACQDEFAULT parameter in the *OPTIONS section of the ODF.
CALENDAR=xxxxxxxx	Points to a calendar used for time-activated Auto Connect sessions.
DELAY=nnnn | 0	Specifies the number of seconds, 0–9999, for Connect:Enterprise to delay after ending one session and before beginning another session with a remote site in the Auto Connect list. The purpose of the DELAY parameter is to allow enough time for the modems to reset between multi-remote Auto Connect sessions. A value of 3–5 seconds is usually sufficient. If a DELAY value is specified, a nonzero DISCINTV value must also be used for this Auto Connect list.

Parameter	Description
DISCINTV=NO | nnnn | 0	Specifies a disconnect interval. If there is no session activity for the number of specified seconds, Connect:Enterprise forces the session to end. This is a safety feature; use it to prevent an Auto Connect session from suspending if a remote site does not respond. The disconnect interval is activated at three specific times: ◆ Immediately after the signon to JES, when first attempting to send to JES ◆ After the send to JES is complete, when a line turnaround to receive is attempted ◆ After a complete batch is received from JES, to allow the receipt of multiple batches from JES When a disconnect interval is used, expect to see time-out messages from BTAM on the system console. The following options are available: ◆ NO—Default. Indicates that no disconnect interval processing is done. Connect:Enterprise uses the standard line time-out as the maximum time to wait for JES to respond. ◆ 0—Specifies an unlimited disconnect interval for Connect:Enterprise. Connect:Enterprise does not disconnect from JES during the three situations when a disconnect interval is activated. This value is not recommended unless another method exists for ending the connection, such as a disconnect from JES. ◆ nnnn—Specify from 1 to 3600 seconds as the disconnect interval.
JES=NO | YES	Specifies whether the remote site is a JES2 site. Connect:Enterprise automatically receives multiple data collections or batches from JES. Specify the BCHSEP=NO parameter on the remote site specification record. This ensures that Connect:Enterprise automatically concatenates multiple batches into a single transmission. Support for the common JES print and punch device select characters is provided. The JES and POWER parameters are mutually exclusive. The default is NO. ◆ NO—Indicates the remote site is not JES2. ◆ YES—Indicates the remote site is JES2. When YES is specified, each remote specification record for this Auto Connect list is immediately followed by the proper JES signon record.

Parameter	Description
LINES=xxxxxxxx | [,xxxxxxxx]	Specifies one or more lines that Connect:Enterprise uses to process an Auto Connect request. Specify the line ID from an auto dial in the User Assembly. Do not specify duplicate IDs in a list. If this parameter is omitted, Connect:Enterprise scans all autodial lines for a single usable line as each Auto Connect session begins. If TYPE=BSCAD, this parameter is optional. If TYPE=BSCMD, this parameter is required. Specify the one manual dial line for the Auto Connect session by using the line ID from a manual dial M$LINEX macro. If TYPE=BSCNS, this parameter is invalid. The line used for the Auto Connect session is based on the remote site specification record for nonswitched lines.
NOBATCH=<u>C</u> | NC	Specifies whether Connect:Enterprise should attempt a connection with a remote site when no batches are ready for transmission. ◆ C—Indicates make a connection regardless. (default) ◆ NC—Indicates no connection when no batches are transmittable. To receive batches from a remote site during an Auto Connect session, specify NOBATCH=C. A connection is made even when no batches are flagged for transmission to the remote site.
POWER=<u>NO</u> | YES	Specifies whether the remote site is a POWER site. Connect:Enterprise automatically receives multiple data collections or batches from POWER. Specify the BCHSEP=NO parameter on the remote site specification record. This ensures that Connect:Enterprise automatically concatenates multiple batches into a single transmission. ◆ NO—The remote site is not POWER. ◆ YES—The remote site is POWER. When YES is specified, each remote specification record for this Auto Connect list is followed immediately by the proper POWER signon record. The JES and POWER parameters are mutually exclusive.
RETRY=<u>0</u> | nn	Specifies the number of times Connect:Enterprise retries any communication failure. If this parameter is used, it is a numeric value 1–99. This retry value is in addition to any retries already done by BTAM. An excessive number of retries can slow down the Auto Connect process if remote sites do not respond promptly to a call.
SIGNOFF=<u>NO</u> | YES	Specifies whether the standard signoff is sent to JES/POWER before the JES/POWER connection is ended. ◆ NO—The signoff is not sent to JES/POWER. ◆ YES—The signoff is sent before ending the connection.

Parameter	Description
TIME=hh:mm | [,hh:mm, ...]	Specifies one or more time-of-day values when Connect:Enterprise automatically activates a full Auto Connect session for the list. Specified as a 4-digit number separated by a colon, and a valid time using a 24-hour clock (for example, 08:00, 14:30). If this parameter is omitted, the only way to activate this list is to use the $$CONNECT console command or the CICS/ISPF interfaces. If the same time is specified for multiple Auto Connect sessions, they are spaced five seconds apart.

Add a BSC Remote Site to a BSC Auto Connect List

Following the Auto Connect session parameters, you must provide one or more remote site specification records. These records list each remote site accessed and additional options for each remote site. A single remote site can be included in many different Auto Connect lists, using different modes and other options.

Remote sites are specified for each Auto Connect list by remote name. This remote name is sometimes used as the mailbox ID of batches sent to the remote site during a fully automated Auto Connect session. The telephone number of each remote site is required for all switched remote sites. Each of these numbers is dialed by the auto dial unit for switched auto dial lines, or is displayed on the console for switched manual dial lines for manually dialing.

A fully automated Auto Connect session can use auto-dial lines or nonswitched (leased) lines. A fully automated Auto Connect session using auto-dial lines proceeds through the list of defined remote sites with the auto-dial unit dialing each telephone number when needed. A fully automated Auto Connect session using nonswitched lines accesses each remote site sequentially until all listed remote sites have been accessed.

You initiate manual dialing through the console command $$CONNECT, the CICS interface, or the ISPF interface.

Note: For manually dialed lines, Connect:Enterprise requires one remote site per Auto Connect list. This ensures that the Auto Connect session does not stall while waiting for access to more than one remote site in the list. Specifying more than one remote site definition record per Auto Connect list that is manually dialed line returns an ODF error, and Connect:Enterprise for z/OS initialization fails.

You must define a value for the MODE parameter for each BSC remote site specification record. The MODE parameter controls how Connect:Enterprise reacts when communicating with the remote site. Connect:Enterprise cannot force a remote site to react properly to the value specified for the MODE parameter; therefore, the value specified for the MODE parameter must be compatible with the capabilities of the remote site equipment. The following modes are allowed, any one of which can be used for different remote sites contained in a single list:

- Send first then receive (SENDRECV)
- Send only (SENDONLY)

- Receive first then send (RECVSEND)
- Receive only (RECVONLY)

For example, to support a 3780 type device that usually first sends data to Connect:Enterprise and then receives data from Connect:Enterprise, the SENDRECV mode is used.

RECVONLY or SENDRECV modes cannot be used on nonswitched lines, because the initial receive will not time out if the remote site does not respond and the Auto Connect session will stall. This restriction does not apply to switched lines.

If you want to temporarily change the communication method with a remote site, you can override the MODE parameter defined in the ODF by using the $$CONNECT console command, the ISPF interface, or the CICS interface.

Separate Connect:Enterprise data transmission options can be specified for individual remote sites in an Auto Connect list. Optimize transmissions by requesting Connect:Enterprise to block and compress data transmissions to the remote site (BLOCK=nn and CMP=Y). If a remote site typically operates using BSC transparency for all transmissions, specify TRANSPAR=Y for that remote site (RECSEP=1E or 1F only). When BTAM ID verification is used, RID supplies the ID to be verified.

If the IDLIST parameter is specified for a remote site, then only batches with the indicated mailbox IDs are transmitted to that site. Otherwise, any batches that match the LISTNAME and remote name are transmitted. You can also use a specific mailbox ID and BATCHID with the $$CONNECT command, the CICS interface, the ISPF interface, or a user-written API program to control what batches are sent to a specific site.

BSC Remote Specification Record Format and Rules

The following example illustrates the format of the BSC remote site specification record. Default values for parameters are underlined.

```
REMOTE_NAME            dd nn...nn | Dnn...nn | CRNnn...nn
                       BCHSEP=NO | OPT1 | OPT2 | OPT3
                       BLOCK=nn | *nn
                       CMP=N|Y
                       HID=xxx...xxx
                       LINEID=xxxxxxx
                       MODE=SENDRECV | SENDONLY | RECVSEND | RECVONLY
                       ONEBATCH=YES | NO
                       RECSEP=1E | 1F
                       RID=xxx...xxx
                       TRANSPAR=N|Y
                       TRUNC=N|Y
                       BEGINLIST=xxxxxxxx
                       IDLIST=xxxxxxxx
                       ENDLIST=xxxxxxxx
```

There is no change to the format of the remote specification records for JES or POWER. However, the MODE parameter must be SENDONLY or SENDRECV to be compatible with JES2 and POWER, which receive first. Connect:Enterprise always sends the signon first, then sends batches, if any, to JES or POWER. Connect:Enterprise resets the line to receive only if MODE=SENDRECV.

When you define BSC remote site specification records, observe the following rules:

- You must include at least one remote site specification record for an Auto Connect list.
- For a BSC Auto Connect that uses a manually dialed line, specify only a single remote site definition per Auto Connect list to avoid an ODF error that causes initialization to fail.
- REMOTE_NAME is required and must be the first parameter specified in a BSC remote site specification record.
- The dd= parameter must be the second operand, following REMOTE_NAME.
- The MODE= parameter is required.
- The RECVONLY and SENDRECV modes are not valid for nonswitched lines.
- Specify all optional parameters in any order on the same line as REMOTE_NAME; separate them by one or more spaces.
- Each JES or POWER remote specification record must be followed by a single record that contains the exact signon for JES or POWER for that remote site. The record is not checked for validity and can contain any data. A JES signon record must not contain an asterisk (*) in column 1; a POWER signon record must begin with an asterisk in column 2.
- The line containing REMOTE_NAME and optional parameters must precede BEGINLIST, IDLIST, and ENDLIST.
- BEGINLIST, IDLIST, and ENDLIST must be specified as the last parameters in a BSC remote site specification record in the following order: BEGINLIST, IDLIST, ENDLIST.

BSC Remote Site Specification Record Parameters

The following table describes BSC remote site specification record parameters. Required parameters are listed in bold first in the table. With the exception of positional parameters, the remaining parameters are listed in alphabetical order. Acceptable abbreviations for parameters are enclosed in parentheses below the parameter in the following table.

Parameter	Description
REMOTE_NAME	Required. A 1–8 character name specifying the REMOTE_NAME for the remote site. This positional parameter must be the first operand on remote site specification records. It is used as the mailbox ID for batches sent to the remote when IDLIST is not specified. Manual dial Auto Connect lists can contain only a single remote record. This ensures that when the Auto Connect session begins, the operator is present and can dial the single remote site.
dd	Specifies the number of digits in the remote site telephone number for a switched remote site. This positional parameter *must be* the second operand. The number must consist of two digits from 01–40. 00 is reserved for nonswitched remote connections.

Parameter	Description
nn...nn | Dnn...nn | CRNnn...nn	Specifies the telephone number of the remote site for a switched remote site only. This positional parameter *must be* the third operand and consist of numeric digits from 0–9, with no dashes. This is a required parameter for switched remote connections. Do not include this operand for nonswitched remote connections. A colon or other special character embedded in the telephone number provides a pause to obtain a dial tone for an outside line (for example, 9:5551212). To determine what characters are acceptable, refer to your autodial manual. ◆ Dnn...nn = SADL modem—For a SADL modem, the phone number prefix is the character D. ◆ CRNnnn...nnn= V25.bis—For a V25.bis modem, the phone number prefix is CRN.
MODE=SENDRECV | SENDONLY | RECVSEND | RECVONLY **(M=)**	Required. Specifies the mode that Connect:Enterprise uses to communicate with the remote site. You can specify your options using either the long form (MODE=SENDRECV) or the short form (M=SR). However, a combination of the two forms (MODE=SR) is not valid. ◆ SENDRECV (M=SR)—Connect:Enterprise first sends batches to the remote site then resets the connection to receive batches from the remote site. ◆ SENDONLY (M=SO)—Connect:Enterprise sends batches to the remote site then disconnects from the remote site. ◆ RECVSEND (M=RS)—Valid only for manual dial-up connections. Connect:Enterprise first receives batches from the remote site, then resets the connection to send batches to the remote. Not used with any nonswitched remote because an initial receive never times out if the remote has nothing to send. ◆ RECVONLY (M=RO)—Valid only for manual dial-up connections. Connect:Enterprise receives batches from the remote site, then it disconnects from the remote site. Not used with any nonswitched remote site because an initial receive never times out if the remote site has nothing to send. When autodialing is used in receive first mode, an ENQ-EOT (send dial sequence and logon sequence) is sent from the BTAM Read Initial macro at the host site. This may cause ENQ contention with the remote site if the remote site is not set up as a receive first to handle the ENQ-EOT sequence.

Parameter	Description
BCHSEP=NO | OPT1 | OPT2 | OPT3	Specifies the method Connect:Enterprise uses to separate batches sent to remote sites when multiple batches are sent in a single connection. Specify only protocols to which the remote sites can properly respond. The default is NO. ◆ NO—Batches are not separated when NO is used. If multiple batches are sent, they are sent as a single batch. Ensure remote sites for this Auto Connect session can process concatenated batches. ◆ OPT1—Connect:Enterprise uses the common RJE method of separating batches. At the end of each batch, Connect:Enterprise sends EOT (X'37'), reads the response, then sends ENQ to request use of the line. Refer also to the RMDC parameter specified in the ODF *OPTIONS section. ◆ OPT2—Connect:Enterprise separates batches with an ETX (X'03'). ◆ OPT3—Batches are not separated. If multiple batches are sent in a single connection, they are concatenated and sent in a single batch. However, the individual batches are not flagged as transmitted until the entire transmission is successfully completed. Ensure remote sites for this Auto Connect session can process concatenated data batches if this option is chosen. The BCHSEP specification has the following order of priority: 1 Taken from the $$CONNECT operator command or from the CICS and ISPF interfaces. 2 Taken from the BSC remote site parameters in the ODF. 3 Taken from the M$LINEX in the user assembly.
BLOCK=nn | *nn	Used to specify the number of records per block (1–99) during an Auto Connect session SEND in which multiple records separated by control characters are transmitted in a single data block. Maximum=99. *1–*99=Transmit the first record unblocked, for example, when the first record is a signon or control record that must be separated from the data. Connect:Enterprise for z/OS transfers the first record by itself and then attempts to transmit all other records in blocks using the BOLCK value specified. Blocking optimizes the use of the transmission lines and can greatly increase the speed of the transmission. To use this operand, a remote site must be able to process blocked records. If the blocking factor is too large for the line buffer size, a smaller blocking factor is used. Omit this operand for unblocked records.
CMP=N|Y	Specifies the use of BSC blank compression to optimize use of the transmission lines during an Auto Connect session SEND to the remote site. The default is N. ◆ N—Indicates that no blank compression is done. ◆ Y—Requests Connect:Enterprise to compress blanks in the data batch. The remote site must be able to decompress or to process compressed data. If a remote site can process a variable number of records per block and BSC blank compression, specify BLOCK=99 and CMP=Y for maximum line optimization.

Parameter	Description
HID=xxx...xxx	Used only if BTAM ID verification is used and if the line uses HOST IDVER only. (M$LINEX specifies IDVER=HOST.)
LINEID=xxxxxxxx	Used with nonswitched remote connections only. It specifies the line ID from a nonswitched M$LINEX macro in the user assembly. Since nonswitched lines are always associated with a specific remote site, this parameter uniquely identifies the remote site that is accessed for an Auto Connect session. If this parameter is omitted and the remote site is nonswitched, Connect:Enterprise uses the remote name as the line ID. The specified LINEID (or the default remote name) must match the ID of a nonswitched M$LINEX macro in your User Assembly.
ONEBATCH=YES | NO (OB=Y | N)	Specifies whether only the first batch that meets the transmission criteria be sent. ◆ NO—Specifies that all batches matching transmission criteria are sent. ◆ YES—Specifies that only the first batch matching transmission criteria is sent.
RECSEP=1E | 1F	Specifies the BSC record separator used for blocked output during the Auto Connect session. ◆ 1E—Specifies the standard record separator for 3780 type devices. ◆ 1F—Specified for 2780 type devices or other remote devices that require its use.
RID=xxx..xxx	Used only if BTAM ID verification is used on the line during an Auto Connect session. The parameter specifies a 1–8 character remote site ID which must be transmitted by the remote site before BTAM permits a switched line connection. If this parameter is specified, ensure that the line used for the Auto Connect session specifies the IDVER option. Specify IDVER=REMOTE on the M$LINEX macro in the User Assembly. If RID is specified for any remote in the Auto Connect list, it is used for all remote definitions in the list.
TRANSPAR=N | Y (TRN=N)	This parameter specifies the use of BSC transparency during an Auto Connect session SEND to the remote site. BSC transparency transmits nontext data, such as object modules or binary data, over telecommunication lines. The default is N. ◆ N—Indicates standard data transmission. ◆ Y—Specifies that Connect:Enterprise use BSC transparency when sending to the remote site. The remote site must be able to operate in BSC Transparent mode.
TRUNC=N | Y	Specifies the use of trailing blank truncation during Auto Connect session SENDS to the remote site. The default is N. ◆ N—Indicates no trailing blank truncation. ◆ Y—Specifies that Connect:Enterprise truncate trailing blanks from data batches to optimize the use of the transmission lines. The remote site must be able to process truncated data.

Parameter	Description
BEGINLIST=xxxxxxxx	Specifies the first batch sent to a remote. This is in addition to the normal search for batches to transmit. This parameter is valid only when accompanied by IDLIST. If no transmittable batches are found for IDLIST, the BEGINLIST batch is not sent. If specified, place BEGINLIST before the IDLIST parameter in the remote specification record.
IDLIST=xxxxxxxx | [,xxxxxxx,...]	Enables you to specify a list of specific mailbox IDs to transmit to the remote site during the Auto Connect session (if this parameter is omitted, batches that match the LISTNAME and remote name are transmitted). Specify one or more mailbox IDs, separated by commas or blanks. Do not enclose the list in parentheses. If all IDs do not fit on a single record, repeat the IDLIST keyword on subsequent records.
ENDLIST=xxxxxxxx	Valid only when accompanied by IDLIST. Transmittable batches identified by ENDLIST are transmitted after IDLIST batches are sent and only if at least one IDLIST batch was actually transmitted. If specified, place ENDLIST after the last iteration of the IDLIST parameter in the remote specification record. IF IDLIST was specified across multiple input records, specify ENDLIST after the last iteration.

Sample *CONNECT Records for BSC Auto Connect Lists

The following sample *CONNECT records illustrate various ways to create BSC Auto Connect lists.

```
*CONNECT
  LISTNAME=LIST1
    TYPE=BSCAD
    LINES=SWLINE1,SWLINE2
    TIME=02:00,04:00
      BOSTON 11 16175551212 MODE=RECVONLY
      NEWYORK 11 12125551212 MODE=RECVONLY
      ATLANTA 11 14045551212 MODE=RECVONLY
      MIAMI 11 13055551212 MODE=RECVONLY
  LISTNAME=LIST2
    TYPE=BSCMD
    LINES=MDLINE1
      CHICAGO 11 13125551212 MODE=SENDRECV BLOCK=6 CMP=Y
  LISTNAME=LIST3
    TYPE=BSCNS
    RETRY=2
    TIME=08:00,09:00,10:00,11:00,12:00,13:00,14:00
    TIME=15:00,16:00,17:00
      OPER100 00 MODE=SENDONLY BLOCK=9 LINEID=LINE001
      OPER200 00 MODE=SENDONLY BLOCK=9 LINEID=LINE001
      OPER300 00 MODE=SENDONLY BLOCK=9 LINEID=LINE001
  LISTNAME=LIST4
    TYPE=BSCAD
    TIME=03:00
    JES=YES
    SIGNOFF=YES
    DISCINTV=120
      JES01 07 5551212 M=SR BLOCK=5 CMP=Y
/*SIGNON RMT050
  LISTNAME=LIST5
    TYPE=BSCAD
    TIME=03:00
    LINES=ADSD01
      DALLAS 14 CRN12145551212 MODE=SENDRECV
  LISTNAME=LIST6
    TYPE=BSCAD
    TIME=02:00
    POWER=YES
    SIGNOFF=YES
    DISCINTV=120
      POWER05 07 5551222 M=SR
 * ..  SIGNON 3
```

In this example, the six BSC Auto Connect lists accomplish the following:

✦ LISTNAME=LIST1

LIST1 is for BSC auto dial remote sites, with two lines to handle the Auto Connect session volume. No failure retries are necessary. The Auto Connect session is activated automatically at 2:00 a.m. and 4:00 a.m. every day. The list contains four remote sites: Boston, New York, Atlanta, and Miami. Connect:Enterprise calls the remote sites, receives data from them, and then disconnects.

✦ LISTNAME=LIST2

LIST2 is for a BSC manual dial remote site in Chicago. No failure retries are necessary. Connect:Enterprise prompts the operator to call the remote site, sends blocked and compressed data batches to the remote site, receives data from the remote site, then disconnects.

✦ LISTNAME=LIST3

LIST3 is for BSC nonswitched remote sites. Failures are retried twice. The Auto Connect session is activated every hour on the hour between 8:00 a.m. and 5:00 p.m. One of the remote sites is staffed by three operators who receive batches with their own unique IDs. Therefore, it is defined as three remote sites with unique remote names using the same physical line. The line defined in the user assembly is LINE001. Connect:Enterprise sends data batches to the remote sites using nine records per block. No data is collected from the remote sites.

✦ LISTNAME=LIST4

LIST4 is for a BSC autodial to a JES remote site. The Auto Connect session is activated at 3:00 a.m. A standard JES signoff is sent when the connection is ended by Connect:Enterprise. The JES signon sent follows the JES01 remote record and does a signon for RMT050.

✦ LISTNAME=LIST5

LIST5 is for BSC autodial using a V.25 bis modem. No failure retries are necessary. The Auto Connect session is activated at 3:00 a.m. every day. The line is defined in the user assembly with MODEM= and only MODE=SENDONLY or SENDRECV are valid. The LINES= parameter is required.

✦ LISTNAME=LIST6

LIST6 is for a BSC autodial to a POWER remote site. The Auto Connect session is activated at 2:00 a.m. A standard POWER signoff is sent when Connect:Enterprise ends the connection. The POWER signon sent follows the POWER05 remote record.

Individual Remote Processing

You may want to use one Auto Connect list that contains all remote BSC sites in your system. If you create a list that contains all BSC remote sites, you may also want to create a list for each site so that if Auto Connect processing fails for any site, you can retry remote sites individually. To do this, define an Auto Connect list (ALL) containing all BSC remote sites and an Auto Connect list for each remote site. To help keep track of LISTNAME and remote site values, use the remote site name as the LISTNAME for an Auto Connect list that contains a single remote site, as shown in the following sample *CONNECT record for BSC sites.

```
*CONNECT
  LISTNAME=ALL
    TYPE=BSCAD
    TIME=06:00
      MAINST 07 5511111 MODE=RECVSEND
      MAPLEAVE 07 5522222 MODE=RECVSEND
      ELMBLVD 07 5533333 MODE=RECVSEND
  LISTNAME=MAINST
    TYPE=BSCAD
      MAINST 07 5511111 MODE=RECVSEND
  LISTNAME=MAPLEAVE
    TYPE=BSCAD
      MAPLEAVE 07 5522222 MODE=RECVSEND
  LISTNAME=ELMBLVD
    TYPE=BSCAD
      ELMBLVD 07 5533333 MODE=RECVSEND
```

Frequent Host-Initiated Transmissions

The following example shows sample *CONNECT records to use if you frequently send data batches from the host site to remote sites with minimal operator intervention. To implement this type of Auto Connect session, supply numerous TIME values in the Auto Connect list.

```
*CONNECT
  LISTNAME=FREQUENT
    TYPE=BSCAD
    TIME=08:00 08:30 09:00 9:30 10:00 10:30 11:00 11:30
    TIME=12:00 12:30 13:00 13:30 14:00 14:30 15:00 15:30
    TIME=16:00 16:30 17:00
      BRANCH01 07 5551212 MODE=SENDOLY
      BRANCH02 07 5551212 MODE=SENDOLY
      BRANCH03 07 5551212 MODE=SENDOLY
      BRANCH04 07 5551212 MODE=SENDOLY
```

Host-to-Host Communications

The Auto Connect feature can transmit batches between Connect:Enterprise systems running on separate mainframes.

The following example shows Connect:Enterprise-to-Connect:Enterprise connections for BSC Auto Connect sessions between two mainframe computers (CE1 and CE2):

```
     CE1
*CONNECT
  LISTNAME=CE2
    TYPE=BSCAD
    TIME=03:00
     CE2 07 5555555 MODE=SENDONLY IDLIST=PAYROLL2

     CE2
*CONNECT
  LISTNAME=CE1
    TYPE=BSCAD
    TIME=04:00
     CE1 07 4444444 MODE=SENDONLY IDLIST=PAYROLL1
```

At 3:00 a.m., CE2 calls CE1 and sends all batches with the ID=PAYROLL1. At 4:00 a.m., CE1 calls CE2 and sends all batches with the ID=PAYROLL2.

Mixed Remote Types for BSC Connections

If you have auto-dial lines, manual dial lines, and nonswitched lines, you need to use at least three separate Auto Connect lists to define your remote sites, as illustrated in the following example.

```
*CONNECT
  LISTNAME=CE2
    TYPE=BSCAD
      BRANCH01 07 5551111 MODE=SENDONLY BLOCK=9
      BRANCH02 07 5552222 MODE=SENDONLY BLOCK=9
      BRANCH03 07 5553333 MODE=SENDONLY BLOCK=9
  LISTNAME=LIST2
    LINES=LINE1
    TYPE=BSCMD
      BRANCH01 07 5554444 MODE=SENDONLY BLOCK=9
  LISTNAME=LIST3
    TYPE=BSCNS
      DALLAS 00 MODE=SENDONLY LINEID=LEASE01
```

Remote Sites Requiring BSC SIGNON

If a remote site requires a SIGNON, send it in the first batch during the Auto Connect session. It is not a parameter or option in the Auto Connect list. This can be done in two ways in Connect:Enterprise:

- Set up Auto Connect sessions by remote name and list name with a SIGNON as the first transmittable batch for the remote name.
- Set up sessions by IDLIST with a SIGNON as the BEGINLIST batch or as the first transmittable batch for the first IDLIST value when BEGINLIST is not specified.

Other Connect:Enterprise Sites Requiring BSC SIGNON

A Connect:Enterprise Auto Connect session can supply the BSC SIGNON to another Connect:Enterprise that uses the remote SIGNON feature. The SIGNON must be supplied as the first batch sent in an Auto Connect list.

Special Remote Handling

There is no limit to the number of Auto Connect lists, so you can set up remote sites that require special handling. Also, a remote site can be specified in different Auto Connect lists if you want to access the site with different options.

For example, if your remote sites receive and send data during the day and send electronic mail at night to a printer, use the following sample Auto Connect lists for reference:

```
*CONNECT
  LISTNAME=DAYTIME
    TYPE=BSCAD
      BRANCH01 07 5551212 MODE=RECVSEND
      BRANCH02 07 5551212 MODE=RECVSEND
  LISTNAME=NIGHTIME
    TYPE=BSCAD
      BRANCH01 07 5551212 MODE=SENDONLY TRN=Y
      BRANCH02 07 5551212 MODE=SENDONLY TRN=Y
```

Sample BSC Options Definition Files

This section provides samples of ODFs for BSC connections.

Simple BSC Connection

The following example shows a simple BTAM connection. A password is defined to allow remote sites to have access to restricted functions, but all other *OPTIONS parameters use default values. No Auto Connect function and no system security are used. The user assembly load module is named MY$USER.

```
*OPTIONS
  BTAM=YES
  VPF='ENTPRS.VPF'
  PASSWORD=BANANA
  UA=MY$USER
```

Complex BSC Connection

The following example shows a more complex system using BTAM, with a password, 10 mailbox IDs for batch security, the console log facility, and mailbox IDs stored in the core to improve efficiency. The Auto Connect feature can send to two of the remote sites. A user-supplied security exit is invoked before Connect:Enterprise performs its standard security checks. For more information on configuring BSC Auto Connects, see *Configuring the *CONNECT Record for BSC Auto Connect Lists* on page 184.

```
*OPTIONS
  BTAM=YES
  VPF='ENTPRS.VPF'
  PASSWORD=AVOCADO
  XSECUR1=STSEC1
  SECURITY=BATCH
  SCINCOR=YES
  CONSLOG=YES
  UA=MY$USER
*SECURITY
  ID=FRESNO ID=LONEPINE ID=TAHOE ID=TERM0001
  ID=TERM0002 ID=TERM0003 ID=TERM0004
  ID=TERM0005 ID=BRANCH1 ID=BRANCH2
*CONNECT
  LISTNAME=AUTOCALL
    TYPE=BSCAD
    RETRY=1
      TERM0001 07 5551212 MODE=SENDONLY
      TERM0002 07 5551212 MODE=SENDONLY
```

Chapter 7

Configuring *CALENDAR Records

Fully automated Auto Connect sessions are activated daily when the system clock reaches the time of day specified in an Auto Connect list. If Connect:Enterprise remains up for multiple days, the Auto Connect session is activated every day when the system clock reaches the specified time. *CALENDAR records enable you to specify dates and days of the week on which to activate or deactivate Auto Connect processing. You apply the schedule specified in a *CALENDAR using the CALENDAR parameter in the *CONNECT record.

Configuring the *CALENDAR Record

*CALENDAR records define dates or days used for time-initiated Auto Connect sessions. Each calendar can list a schedule of days, dates, or both that specify when to activate or deactivate an Auto Connect session. Calendars are indicated by a *CALENDAR record. You can use the *CALENDAR record with the TIME parameter, which specifies daily processing times for Auto Connects, to refine Auto Connect scheduling.

*CALENDAR Record Format and Rules

The following table shows the *CALENDAR record format:

```
*CALENDAR              NAME=xxxxxxxx
                       DATES=mm/dd
                       EXDATES=mm/dd
                       EXDAYS=SUN MON TUE WED THU FRI SAT
```

When you define the *CALENDAR record, observe the following rules:

- *CALENDAR must begin in column 1. Any other text on the same line is ignored.
- The *CALENDAR record is followed by one or more calendar definitions, each beginning with the NAME= keyword.

- The NAME= keyword can begin in any column. Any other text on the same line is ignored.
- Each NAME= record is followed by one or more keyword parameters that supply days, dates, or both to activate or bypass the Auto Connect function. The keywords begin in any column, can be in any order, and must always directly follow the NAME record.

*CALENDAR Record Parameters

The following table describes the *CALENDAR record parameters.

Parameter	Description
NAME=xxxxxxxx	Specifies the name identifying the calendar. Each calendar defined must have a unique name.
DATES=mm/dd	Specifies any dates on which to activate the Auto Connect function. Multiple dates (mm/dd) specified on a single record are separated by blanks or commas. The values for DATES= *cannot be* the same as EXDATES=.
EXDATES=mm/dd	Specifies any dates on which to bypass the Auto Connect function (EXception DATES). Multiple dates (MM./dd) specified on a single record are separated by blanks or commas. The values for EXDATES *cannot be* the same as the values for DATES.
EXDAYS=SUN MON TUE WED THU FRI SAT	Specifies any days of the week on which to bypass the Auto Connect function (EXception DAYS). Days which are not specified with this keyword default to activation days. Multiple days specified on a single record *must be* separated by blanks or commas.

Auto Connect *CALENDAR Processing Rules

The processing rules for *CALENDAR records are as follows:

- Each specified calendar date and each day of the week is associated with an action indicator: to activate or not to activate.
- Dates override days of the week.
- If a calendar date entry matches the current date, the associated action is performed. If the current date is not specified in the calendar, the action associated with the current day of the week is performed.
- By default, each day is flagged for activation unless it is identified as an exception day (EXDAYS).

Connect:Enterprise processes calendar values in the following order:

1. Connect:Enterprise checks the date. If a match is found against the current date, perform the specified action.
2. Connect:Enterprise checks the day of the week. If no date match is found in the previous step, perform the specified action for the current day of the week.

Sample *CALENDAR Record

The following sample *CALENDAR record defines six schedules that activate and deactivate Auto Connect processing. You can apply these schedules to Auto Connect lists by defining the CALENDAR= parameter in the *CONNECT record.

```
*CALENDAR
  NAME=SCHED1
    EXDAYS=MON TUE WED THU FRI
  NAME=SCHED2
    EXDAYS=SAT SUN
    DATES=07/04
  NAME=SCHED3
    EXDAYS=SUN
  NAME=SCHED4
    EXDATES=05/30 07/04 12/25
  NAME=SCHED5
    EXDAYS=SUN MON TUE WED THU
    EXDATES=01/01 07/04 12/25
  NAME=SCHED6
    EXDAYS=SUN MON TUE WED THU FRI SAT
    DATES=01/01,01/15,02/01,02/15,03/01,03/15,04/01,04/15
    DATES=05/01,05/15,06/01,06/15,07/01,07/15,08/01,08/15
    DATES=09/01,09/15,10/01,10/15,11/01,11/15,12/01,12/15
```

The sample *CALENDAR records specify the following processing:

- Calendar SCHED1 bypasses the Auto Connect function Monday through Friday.
- Calendar SCHED2 bypasses the Auto Connect function Saturdays and Sundays. Additionally, the Auto Connect function is activated on 07/04, regardless of which day of the week this date falls on.
- Calendar SCHED3 bypasses the Auto Connect function every Sunday.
- Calendar SCHED4 bypasses the Auto Connect function on 05/30, 07/04, and 12/25.
- Calendar SCHED5 bypasses the Auto Connect function Sunday through Thursday. Additionally, the Auto Connect function is bypassed on 01/01, 07/04, and 12/25 regardless of what day of the week these dates fall on.
- Calendar SCHED6 activates the Auto Connect function on the 1st and 15th of each month.

Chapter 8

Creating the Connect:Enterprise Startup Task

After unloading the Connect:Enterprise release tape, setting up and starting the VSAM file server, and creating your site-specific ODF, you are ready to create the Connect:Enterprise startup task by editing and running the startup JCL. This procedure also verifies your ODF.

Editing the Connect:Enterprise Startup JCL

This procedure assumes that you have started the VSAM file server and configured your site-specific ODF. After you complete these tasks, start Connect:Enterprise:

1. Copy JCL EXAMPLE member ENTPRS to a JES defined PROCLIB by a name of your choice. You can assign any name. This will be the Connect:Enterprise startup JCL.
2. Edit the renamed JCL.
3. Add ODF parameters after *OPTIONS in the first STEP or remove optional STEP 1 if the ODF is already defined.
4. To start Connect:Enterprise with the ODF Verify option:
    a. Change the following lines to comment the line that contains only PROC NAME=SRV1, and uncomment the line that contains PROC NAME=SRV1 and VERIFY=VERIFYONLY:

    ```
    //ENTPRS   PROC NAME=SRV1
    //*ENTPRS  PROC NAME=SRV1,VERIFY=VERIFYONLY
    ```

    b. Change the following lines to comment the line that contains only PARM=’&NAME’, and uncomment the line that contains PARM=’&NAME,&VERIFY’:

```
//          PARM='&NAME'
//*         PARM='&NAME,&VERIFY'
```

The VERIFYONLY option scans the ODF, returns error messages, and terminates Connect:Enterprise for z/OS whether or not ODF errors are detected.

5. Replace ENTPRS.LOAD with your load library name.
6. Replace ENTPRS.OPTFILE with your ODF name.
7. Replace ENTPRS.SNAPOUT with your snapshot data set name. If you prefer, the snapshot data set can be written directly to SYSOUT. Specify DCB information as:

```
//SNAPOUT DD   SYSOUT=*,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
```

8. Replace ENTPRS.BTSNAP with your ESTAE DUMPS data set name. The BTSNAP data set can be written directly to SYSOUT. Specify DCB information as:

```
//BTSNAP DD   SYSOUT=*,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
```

9. When an FTP configuration is defined (FTP=YES), replace ENTPRS.EXAMPLE (TZ) with your specific environment variable.
10. Replace ENTPRS.SCRIPT with your FTP script library name.
11. Replace ENTPRS.RULES, ENTPRS.RULES.CNTL, and ENTPRS.RULES.TRACE with your rules libraries.
12. For BSC sites, insert the proper line ID DD statements for your Connect:Enterprise configuration. See *BTAM Line Considerations (BSC Only)* on page 209 and *Connect:Enterprise for z/OS for z/OS Installation Guide*, for more information about BSC configuration.
13. The JESRDR DD is required for the Connect:Enterprise job submission feature.
14. Submit the JCL to start online Connect:Enterprise.

    The following messages are displayed on the system console as Connect:Enterprise starts:

```
CMB333I - Connect:Enterprise GLOBAL STORAGE BLOCK BUILT AT: 0A5CD000 FOR A LENGTH OF: 00003000
CMB277I - ODF VERIFICATION STARTED
CMB278I - ODF VERIFICATION COMPLETE, 000000 ERRORS FOUND IN ODF PARAMETERS
CMB219I - CURRENT COLLECTION FILES ARE VBQxx AND VLFx
CMB170I - MAXIMUM BATCHES nnnnnnnn, CURRENT BATCHES nnnnnnn, LAST USED nnnnnnn, ROLLED nnnnnnn
CMB002I - Connect:Enterprise Connect:Enterprise V01.R02.M00 INITIALIZATION COMPLETE
CMB171I - Connect:Enterprise Connect:Enterprise NOW USING MODIFY INTERFACE
CMB096I - Connect:Enterprise Connect:Enterprise/SNA VTAM ACB OPEN
CMB2101I - Connect:Enterprise/TCPS TCP/IP FEATURE INITIALIZATION IN PROGRESS.
CMB2108I - Connect:Enterprise/TCPS TCP/IP C ENVIRONMENT MANAGER TASK INITIALIZATION COMPLETE.
CMB2102I - Connect:Enterprise/TCPS TCP/IP THREAD INITIALIZATION IN PROGRESS.
CMB2109I - Connect:Enterprise/TCPS TCP/IP FTP LISTENER INITIALIZATION COMPLETE.
CMB2135I - Connect:Enterprise/TCPS TCP/IP AUTO Connect MANAGER INITIALIZATION COMPLETE.
CMB2103I - Connect:Enterprise/TCPS TCP/IP FTP SERVER THREAD INITIALIZATION COMPLETE. nnnn SESSION THREADS
ALLOCATED.
CMB2103I - Connect:Enterprise/TCPS TCP/IP FTP CLIENT THREAD INITIALIZATION COMPLETE. nnnn SESSION THREADS
ALLOCATED.
CMB353I - Connect:Enterprise/APPC VTAM ACB OPEN
CMB124I - Connect:Enterprise/APPC INITIALIZATION COMPLETE
```

If Connect:Enterprise terminates because of an error, the Connect:Enterprise job indicates a completion code of USER=253. This user ABEND code is always accompanied by one or more console messages that further describe the detected error condition.

Online Connect:Enterprise runs until the host operator shuts it down.

VTAM Network Considerations (SNA Only)

You must complete all VTAM initialization and network activation before you execute Connect:Enterprise. The ENTPRS JCL does not contain DD statements for VTAM lines because they are controlled by VTAM. Do not allocate these lines to Connect:Enterprise by using the ENTPRS JCL.

BTAM Line Considerations (BSC Only)

BTAM lines defined in the user assembly are accessed at Connect:Enterprise startup and must be allocated to the Connect:Enterprise job.

Any lines defined by M$LINEX must be allocated in the ENTPRS JCL. Each line is allocated to the job in a DD statement. Each defined line is associated with a physical unit address, which can be obtained from your installation's systems programmer.

For example, to assign three BSC switched lines and one BSC nonswitched line to Connect:Enterprise, your systems programmer generated the operating system with BSC switched lines 012, 013, and 014 and BSC nonswitched line 028 for use by Connect:Enterprise. A user assembly is generated with the following M$LINEX macros:

```
M$LINEX ID=LINE01,TYPE=BSCSW
M$LINEX ID=LINE02,TYPE=BSCSW
M$LINEX ID=LINE03,TYPE=BSCSW,DIALOUT=AUTO
M$LINEX ID=BRANCH01,TYPE=BSCNS,BUFSIZ=2000
M$ENDX
END
```

The DD statements to allocate these lines to Connect:Enterprise are shown in the following example:

```
//LINE01 DD UNIT=012
//LINE02 DD UNIT=013
//LINE03 DD UNIT=014
//BRANCH01 DD UNIT=028
```

Chapter 9

FTP Auto Connect Scripts

This chapter describes FTP Auto Connect scripting using the REXX language, script commands, and how to use variables in REXX scripts. For a general overview of Auto Connects and how to configure ODF records for FTP Auto Connects, see Chapter 5, *Configuring ODF Records for FTP Connections*.

About FTP Auto Connect Scripts

FTP Auto Connect sessions provide an interface with a remote FTP server implemented on any platform. FTP Auto Connects differ from SNA and BSC Auto Connects in that FTP Auto Connects use REXX language scripts to control both the connection to the remote server and data transmission between the client and remote server. These scripts execute automatically. They can be passed variables, which allow the scripts to be reused for different sessions.

FTP Auto Connects use two scripts: LOGON_SCRIPT and the AC_SCRIPT. These scripts are members in PDS files on the SYSEXEC DD.

The LOGON_SCRIPT

The LOGON_SCRIPT typically controls the logon connection to the FTP server. The connection information includes:

- IP address or domain name
- Port number
- User ID
- Password
- Account value (optional)
- Site commands (optional)
- Firewall negotiation information

You can specify the values for this information directly in the script, or through script variables in the Options Definition File (ODF).

You must have a LOGON_SCRIPT (although it can be blank–see Chapter 5, *Configuring ODF Records for FTP Connections*). A LOGON_SCRIPT can be assigned to each ODF *REMOTES FTP_SERVER entry. If a LOGON_SCRIPT is not specified on the *REMOTES definition for a particular remote, the logon script specified by the ODF *OPTIONS FTP_LOGON_SCRIPT_DEFAULT parameter is used. If the LOGON_SCRIPT is not specified in either ODF record, an error occurs.

See *LOGON_SCRIPT RDXFTPAC Command Summary* on page 213 for details about the LOGON_SCRIPT.

The AC_SCRIPT

The AC_SCRIPT contains REXX language instructions and host environment commands that define the interaction with remote FTP servers. The host environment commands update local client parameters or generate FTP commands issued to a remote FTP server. Some script commands generate several FTP commands.

An AC_SCRIPT is assigned to each ODF *CONNECT FTP remote specification record. If the AC_SCRIPT is not specified on the *CONNECT remote specification record, the AC script specified by the FTP_AC_SCRIPT_DEFAULT *OPTIONS ODF parameter is used. If it not specified in either ODF record, only the LOGON_SCRIPT is executed.

The $$CONNECT console command will override the AC_SCRIPT, but not the LOGON_SCRIPT.

See *AC_SCRIPT RDXFTPAC Command Summary* on page 214 for details about the AC_SCRIPT.

See *Example with Sample Scripts and ODF* on page 222 for more information about sample REXX scripts.

ODF Setup for FTP Auto Connects

To set up an FTP Auto Connect, you must create ODF *CONNECT and *REMOTES entries that specify the FTP server. See Chapter 5, *Configuring ODF Records for FTP Connections*, for instructions and sample ODF records.

Each defined remote FTP server uses a set of variables that can be specified in the ODF. You can also use a script to temporarily modify these values during the script's execution.

Note: A remote name can be defined as both an FTP_CLIENT and an FTP_SERVER. See Chapter 11, *Setting Up Connections to Other Communications Products*, for more information.

The REXX Language

The LOGON_SCRIPT and AC_SCRIPT use Restructured Extended Executor (REXX) language. The REXX language is fully supported in IBM for z/OS and integrated with Connect:Enterprise . REXX supports script syntax checking and process flow. REXX scripts enable calls to other programs and sub-routines, providing a high degree of customization and flexibility.

Because Connect:Enterprise uses REXX for FTP scripting, it is important to understand basic REXX scripting language syntax. The following sections describe REXX highlights that pertain to Connect:Enterprise scripts. See the REXX documentation for a full understanding of the REXX language.

REXX Host Command Environments

REXX consists of the REXX language processor and a number of Host Command Environments (HCE). A REXX script consists of expressions that are evaluated by the REXX language processor. REXX substitutes any variables found in the expression, performs any operation specified (add, divide, for example) and performs any function calls specified. If the REXX language processor encounters an expression in the script that it does not recognize as a REXX keyword instruction or as an assignment instruction, it considers the expression to be a host command and routes it to the current HCE for processing. The HCE processes the command and then returns control to the language processor. The language processor also receives return codes from the HCE that the REXX instructions can use to control the script flow.

When Connect:Enterprise is started, it creates a default REXX HCE named RDXFTPAC. All expressions that REXX does not recognize as REXX instructions in the LOGON_SCRIPT or AC_SCRIPT are passed to the RDXFTPAC HCE.

REXX evaluates all expressions before passing them to the HCE. If you need to pass an expression to the RDXFTPAC HCE and you do not want REXX to evaluate it first, enclose it in single or double quotes. In the following example, the host command PASS is passed unchanged to the RDXFTPAC HCE because it is in double quotes. However, because PASSWORD is not in quotes, REXX considers it a variable and substitutes the value from the ODF *REMOTES &PASSWORD definition for it before passing the expression to RDXFTPAC. (The value for PASSWORD can also be declared in the script.) See *Using Variables in Scripts* on page 216 for more information about variables.

```
“PASS” PASSWORD
```

A REXX variable defaults to the upper case version of its name if it is not initialized previously in the script.

HCE RDXFTPAC Host Commands

HCE RDXFTPAC host commands either update local client parameters, generate FTP commands that are issued to a remote FTP server, or perform special processing.

The following sections summarize RDXFTPAC host commands. Although commands are listed under the script they are typically used in, they can be used in either the LOGON_SCRIPT or AC_SCRIPT.

LOGON_SCRIPT RDXFTPAC Command Summary

The following table summarizes the RDXFTPAC commands typically used in a LOGON_SCRIPT. For detailed explanations of these commands, see *Script Command Details* on page 228.

These commands are not case-sensitive. Case settings are kept for all values within quotes and all values assigned through variable substitution. Any value not in quotes or not through variable substitution is converted to uppercase.

Command	Description
ACCT or ACC	Provides account information to the remote FTP server.
OPEN or O	Opens a connection with a remote FTP server or proxy firewall.
PASS or PA	Sends a password to the remote FTP server or firewall. It also can be used to change a password. Note that passwords are often case sensitive.
SITE or SI	Sends site-specific information to the remote FTP server.
USER or U	Sends an identifying name to the remote FTP server.
USERLOG or UL	Writes a log record with the specified text and user-defined fail codes (240-255).

AC_SCRIPT RDXFTPAC Command Summary

The following table summarizes the RDXFTPAC local commands and the FTP client commands typically used in an AC_SCRIPT. Local session commands modify characteristics of the session while FTP client commands are translated into FTP server commands and are sent to the remote FTP server. Connect:Enterprise FTP Client command implementation requires active connection between Connect:Enterprise and a remote FTP server that complies with RFC 959 and RFC 1123.

For detailed explanations of these commands, see *Script Command Details* on page 228.

These commands are not case-sensitive. Case settings are kept for all values within quotes and all values assigned through variable substitution. Any value not in quotes or not through variable substitution is converted to uppercase.

Host Command	Description
ASCII, ASC, or AS	Sets transfer type to ASCII
BINARY, BIN, or B	Sets transfer type to IMAGE.
CD, CWD, or CW	Changes current working directory at the remote FTP server
CDUP	Changes to parent directory at the remote FTP server.
DELETE or DELE	Deletes one file on the remote FTP server.
DIR or DI	Lists the remote FTP server’s file directory.
EBCDIC, EBC, or EB	Sets transfer type to EBCDIC.
GET	Retrieves one file from the remote FTP server to the local host.
IMAGE	Sets transfer type to Image.
LIST or LI	Lists the remote FTP server’s file directory.

Host Command	Description
LOCCD	Changes local current working directory (Mailbox ID).
LOCDIR	Lists the contents of the specified local current working directory (Mailbox ID).
LOCPWD	Displays the local current working directory (Mailbox ID).
LOCSITE or LOCSI	Changes the local SITE parameter values.
LOCSTAT or LOCST	Displays the local SITE parameter values.
LS	Lists the remote FTP server’s file names from the current or specified directory.
MDELETE or MD	Deletes one or more files on the remote FTP server.
MGET or MG	Retrieves multiple files from the remote FTP server to the local host.
MKDIR, MKD, or MK	Creates a directory on the remote FTP server.
MODE or MO	Sets transfer mode–Blocked, Compressed, or Stream.
MPUT or MP	Copies multiple files from the local host to the remote FTP server.
NLST or NL	Names list. See the LS command.
NOOP	Requests a positive response from the remote FTP server.
PUT or PU	Transfers batches from the local host to the remote FTP server.
PWD	Displays the name of the remote FTP server current working directory.
QUIT or QUI	Disconnects from the remote FTP server and ends the session.
QUOTE or QUO	Sends an uninterpreted string to the remote FTP server.
RETR or RET	Retrieves one file from the remote FTP server to a Connect:Enterprise batch.
RMDIR, RMD, or RM	Removes a directory from the remote FTP server.
SCGET	Retrieves one or more batches from a remote Connect:Enterprise for z/OS directory.
SCPUT	Sends one or more batches to a remote Connect:Enterprise for z/OS directory.
SITE or SI	Sends site-specific information to the remote FTP server.
STATUS, STAT, or STA	Retrieves status information from the remote FTP server.
STOR or STO	Transfers a file from the local Connect:Enterprise to the remote FTP server and names the file.
STOU	Transfers a file from the local Connect:Enterprise to the remote FTP server without naming the file.
STRU or STR	Sets transfer file structure as File or Record.
SUNIQUE or SU	Selects either STOR or STOU for PUT and MPUT commands.
SYSTEM, SYST, or SY	Requests name and level of the remote FTP server operating system.

Host Command	Description
TYPE or TY	Sets transfer type as ASCII, EBCDIC, or Image.
USERLOG or UL	Writes log record that includes user specified text and user-defined fail codes (240-255).
!TIMER	Turns off or on the exec loop/hang timer.
!WTO	Writes a string of text to the system using WTO Routcde=11, DESC=7.

Using Variables in Scripts

You can use variables to create generic scripts that communicate with several different FTP servers.

Some script variables are preset in the ODF. They are set for both the LOGON_SCRIPT and AC_SCRIPT programs, but are also set for any nested sub-programs. These variables can be modified within a script, but the modifications are lost when the script ends. Variable modifications made in the LOGON_SCRIPT are not passed to the AC_SCRIPT.

The following table lists the variables that can be preset for a REXX language script and their sources:

ODF Variable Name	REXX Variable Name	Source	Possible Use	Default Value
&BEGINLIST	BEGINLIST	*CONNECT	Mailbox ID	
&BID	BID	*REMOTE	User Batch ID	‘NONE’
		$$CONNECT	User Batch ID or Batch Number	‘NONE’
&DATAMODE	DATAMODE	*REMOTE	Transmission mode	Stream
&DATASTRU	DATASTRU	*REMOTE	Data structure	File
&DATATYPE	DATATYPE	*REMOTE	Data type	ASCII
&ENDLIST	ENDLIST	*CONNECT	Mailbox ID	
&IDLIST	IDLIST	*CONNECT	Mailbox ID	
&IPADDR	IPADDR	*REMOTE	IP Address	
&NEWPASS	NEWPASS	*REMOTE	New password	
&PASSWORD	PASSWORD	*REMOTE	Password	
&PORTNO	PORTNO	*REMOTE	Port number of remote server	0021
&RECVPATH	RECVPATH	*REMOTE	Directory path	
&SENDPATH	SENDPATH	*REMOTE	Directory path	
&USERID	USERID	*REMOTE	User	Remote Name

Variables definitions in the ODF are preceded by an ampersand as shown in the following example of script variables in the ODF *REMOTES FTP_SERVER definition:

```
*REMOTES
     NAME=FTPRMT1
     TYPE=FTP_SERVER
     &IPADDR=10.20.139.44
     &PORTNO=5546
     &USERID=GENUSR1
     &PASSWORD=MYPASS44
```

Variables specified in REXX scripts do not use the ampersand prefix.

There are also preset variables available to both scripts that cannot be specified in the ODF or modified by the script or by the CICS or ISPF interface. The following table lists these additional variables and the values assigned by Connect:Enterprise :

Variable Name	Connect:Enterprise Assigned Use
BATCH#	The 1–7 digit number (including any leading zeros and optionally preceded by the # sign) of the batch being processed at the time it is resolved. Used only if the $$CON command used to start the auto connect session contained both the ID and BID parameters, and the BID parameter used valid batch number syntax.
BATCHES_RECEIVED	The 1-7 digit count of batches received so far in the FTP session (both LOGON and AC scripts).
BATCHES_SENT	The 1-7 digit count of batches sent so far in the FTP session (both LOGON and AC scripts).
BYTES_RECEIVED	The 1-11 digit count of bytes received so far in the FTP session (both LOGON and AC scripts).
BYTES_SENT	The 1-11 digit count of bytes sent so far in the FTP session (both LOGON and AC scripts).
CLIENT_CNTL_ IPADDR	The IP address of the last established FTP client control connection (nnn.nnn.nnn.nnn).
CLIENT_CNTL_ PORTNO	The port number of the last established FTP client control connection (0-65535).
CLIENT_DATA_ IPADDR	The IP address of the last established FTP client data connection (nnn.nnn.nnn.nnn).
CLIENT_DATA_ PORTNO	The port number of the last established FTP client data connection (0-65535).
CNT	Specifies a 1–3 digit number, starting with 0 and incremented by one for each STOR or STOU command sent to the remote FTP server during a given session.
DATE	Provides the Julian date when the session started in yyddd format.
DIR	The stem of the compound variables that holds the content messages returned by the DIR host command.

Variable Name	Connect:Enterprise Assigned Use
DTDDNAME	The DDNAME of the FTP client session's dialog trace DD if dialog trace is active for the remote; otherwise, blank.
EXECNEST	The nesting level of the exec, “1” for the LOGON_SCRIPT or AC_SCRIPT.
HCRC	Host Command Return Code.
LASTRC	A 3-digit response code from the local system or the server.
LISTNAME	The Auto Connect Listname specified in the NAME parameter in the list definition.
LOCDIR	Stem of the compound variables that hold the content messages returned by the LOCDIR host command.
LOCSCAN	The current value of the local (client-side) SCAN setting, which indicates whether the Connect:Enterprise FTP client is scanning RETR received batches for $$ commands and /* cards. This variable is set upon initialization of the client and is updated by the LOCSITE SCAN= command.
ID	The Mailbox ID specified in the $$CONNECT command.
MBXHLQ	The value specified in the *OPTIONS MBXHLQ= parameter (default value is MAILBOX).
MBXNAME	The value specified in the *OPTIONS = MBXNAME parameter (default value is MAILBOX).
MAXRC	The maximum value of LASTRC for the exec.
RDXVARS	Contains a list of all Connect:Enterprise REXX variables except itself, the REPLY., DIR. and LOCDIR. variables. Sample script FTPACVAR uses this variable to display all Connect:Enterprise REXX non-compound variables and their settings.
REPLY	Stem of the compound variables that hold the reply messages to a Host Command
RMTNAME	The remote name from the NAME parameter in the remote definition.
SCAN	The last known value of the remote (server-side) SCAN setting, which indicates whether the Connect:Enterprise FTP server is scanning STOR/STOU received batches for $$ commands and /* cards. Initially, this value is null since it cannot be known until the server is contacted. This variable is set after a STAT command response is processed by the client, or a SITE SCAN= command is issued by the client.
SCRIPTNAME	The name of the LOGON_SCRIPT or AC_SCRIPT running.
SCRIPTTYPE	SCRIPTNAME type – “A” for AC_SCRIPT or “L” for LOGON_SCRIPT.
SERVER_CNTL_ IPADDR	The IP address of the last established FTP server control connection (nnn.nnn.nnn.nnn).
SERVER_CNTL_ PORTNO	The port number of the last established FTP server control connection (0-65535).

Variable Name	Connect:Enterprise Assigned Use
SERVER_DATA_ IPADDR	The IP address of the last established FTP server data connection (nnn.nnn.nnn.nnn).
SERVER_DATA_ PORTNO	The port number of the last established FTP server data connection (0-65535).
SSL_CCC_POLICY	The SSL CCC policy in effect for the FTP session (from SSL_CCC_POLICY in the remote definition, or if not specified there, from SSL_DEFAULT_SERVER_CCC_POLICY).
SSL_POLICY	The SSL policy in effect for the FTP session (from SSL_POLICY in the remote definition, or if not specified there, from SSL_DEFAULT_POLICY).
THREADID	The current thread ID of the FTP client. Value will be FTPC*nnnn* where *nnnn* is the number of the FTP Client thread used to run the Auto Connect.

REXX REPLY. Variables

When an RDXFTPAC host command is executed, it may return one or more response lines to the executing routine. Response text lines are set in REXX compound variables REPLY.1 through REPLY.n. The number of lines is set in the REPLY.0 compound variable. If REPLY.0 is 0, there are no response lines. These variables can be parsed by the executing program commands to determine, for example, the type of FTP server the client is connected to.

Connect:Enterprise FTP Client also puts all its command response messages, which start with a three-digit completion code, into REXX REPLY.n variables (see *Connect:Enterprise for z/OS Messages and Codes Guide* for more information about FTP replies). If you use scripts that check a specific REPLY.*n*, you may need to check more or all of the REPLY.*n* variables generated by a command. Also, because the value of the LASTRC variable changes for some commands since it is taken from the last message put into a REPLY. variable, you may need to make additional changes to scripts.

REPLY. variables, like the DIR. and LOCDIR. compound variables discussed below, are not copied to lower level nested REXX execs. They are only available to the REXX exec that actually issued the command. As with the DIR. and LOCDIR. variables, you cannot issue a command and then call another exec to analyze the reply. Some programming alternatives to using nested REXX execs include:

- Pass replies to the called exec as arguments
- Copy the called exec into the calling exec making it an internal subroutine

REXX LOCDIR. and DIR.Variables

When the RDXFTPAC host commands DIR and LOCDIR are executed, they may return one or more content lines. These lines are not set in the REPLY. variables. They are set in variables DIR.1 through DIR.*n*, and LOCDIR.1 through LOCDIR.*n* (hereafter referred to as "compound variables"), for the DIR and LOCDIR commands respectively. The number of content lines is set in DIR.0 (or LOCDIR.0). If the value of DIR.0 is 0 (zero) all other DIR. (or LOCDIR.) compound variables are not initialized. The compound variables can be parsed by the REXX exec to determine, for example, which of the batches needs to be transmitted based upon any criteria listed in the content line. The compound variables are set regardless of the remote's dialog trace setting. Unlike the fixed variables listed in variable RDXVARS, DIR. and LOCDIR. compound variables are not copied to lower level nested REXX execs. They are only available to the REXX exec that actually issued the DIR or LOCDIR host command. If you need them in a nested REXX exec, you must either issue the host command in the nested exec, or pass the information in an argument to the lower level exec. The DIR (or LOCDIR) host command return code (in variable HCRC) must be 0 before you can rely on the completeness of the DIR. (or LOCDIR.) compound variables.

The EXAMPLE library contains member LOCDIR which shows how the LOCDIR. compound variables can be used with the PUT command to transfer batches which have not already been transmitted to the remote server.

Additional DIR. Variable Information

DIR. compound variables are only created for the DIR host command – not the LIST or NLST host commands. In addition, because they depend on a remote's response, their creation is less certain than the creation of the LOCDIR. compound variables. Upon receiving the first data buffer from the server, the REXX stem "DIR." is dropped, and the variable DIR.0 is set to 0. Then, for each data buffer the data is separated into individual content lines, if possible, and a DIR. compound variable is created for each content line. If it is not possible to determine the separator string, DIR.0 is set to 1, and DIR.1 is set to the entire data response, up to the REXX limit for a variable's length.

When the DIR command returns control to the script, if the variable HCRC is 0, the host command completed successfully and DIR.0 contains a non-zero number indicating the highest DIR. compound variable created. If HCRC is not 0, the DIR. compound variables (including DIR.0) may be unchanged, uninitialized, partially completed, or even complete, depending on the circumstances. For example, if the server does not return any directory data (that is, if it never opened a data port in response to the DIR command), HCRC is set to 4 and the DIR. compound variables are unchanged. The Connect:Enterprise FTP server falls into that category: when there are no matching batches to list, it only sends the reply "550 LIST failed. No batches found." and so the DIR. compound variables are unchanged. The following REXX code example uses the DIR. compound variables.

```
"DIR /MBXID/"
if hcrc = 0 then do
   say "DIR.0 =" dir.0
   do i = 1 to dir.0
      say "DIR."i "=" space(dir.i)
   end
end
```

REXX Literals

A literal string is a character sequence delimited by a single quotation mark (’) or a double quotation mark (“). REXX removes the delimiter before passing the string to the host command. This can complicate how literal strings are resolved in REXX. The following are some suggestions for using quotation marks in a literal string:

- Use two consecutive double quotation marks (“ ”) to represent a double quote character within a string delimited by double quotation marks. For example, BID= “ “ “abc” ” ” resolves to “abc”.
- Use two consecutive single quotation marks (’ ’) to represent a single quote character within a string delimited by single quotation marks. For example, BID= ‘‘‘abc’’’ resolves to ‘abc’.
- If the literal string contains single quotes but no double quotes, delimit it with double quotes. For example, BID=“ ‘abc’ ” resolves to ‘abc’.
- If the literal string contains double quotes but no single quotes, delimit it with single quotes. For example, BID= ‘ “abc” ’resolves to “abc”.
- Create REXX symbols for both single and double quote characters. For example, qq = ‘ “ ’. Then use the symbols to create the final string. For example, BID= qq ‘abc’qq resolves to “abc”.
- Make separate literal tokens out of embedded quotes and use the abuttal operator to concatenate them with other literal tokens. For example, BID= ’ “ ’ || ’abc’ || ’ ” ’ resolves to “abc”.

Script Message Output

REXX writes all messages not destined for the console to ddname SYSTSPRT. If Connect:Enterprise dialog trace is turned on for the remote, the ddname is changed to DTnnnnnn, where nnnnnn starts at 000001 and increments by one for each FTP Auto Connect session. If dialog trace is not turned on for the remote, no ddname is allocated for the REXX output messages and they are discarded. This includes all output from REXX SAY instructions, TRACE output, and IRX* (REXX) error messages.

The special RDXFTPAC host command !WTO enables the script to issue a message that is not lost when dialog trace is off because it writes to both the Job Entry Subsystem (JES) Job Log DD and to the console. However, the Script message output can be suppressed by MPF.

Checking the Results of Script Host Commands

Your scripts should always check the results of each host command issued. Most returned exception conditions do not automatically terminate the script. This means that the HCE will continue to call the script processor and pass host commands to the Auto Client command processor.

The results of a host command are communicated in the Host Command Return Code (HCRC), the Maximum Reply Code (MAXRC) and the Last Reply Code (LASTRC) variables. Check the HCRC first as it describes the overall success or failure of the command and provides system status information. The MAXRC and LASTRC variables contain the 3-digit FTP reply codes returned from both local and remote client FTP commands.

The REXX variable RC is set to the same value as HCRC. However, HCRC is only set by RDXFTPAC host commands, whereas RC can be set by other instructions as well.

The HCRC variable can be set to the following values:

Value	Description
0	Host command successfully processed. All server reply codes were positive replies.
4	Host command successfully processed. One or more negative server or local reply codes were received. MAXRC and LASTRC contain the maximum reply and last reply code received. The server control channel, if established, remains connected.
6	A quiesce shutdown of the FTP feature or the system has been requested. Use the LASTRC and MAXRC variables to check the success or failure of the host command. The FTP session remains connected and additional host commands will be processed.
8	The Open command failed. The connection with the server could not be established.
12	Data transfer command terminated due to data port connection failure. A file transfer that was in-process did not complete. Check the AC log records to determine the state of the batches involved in the transfer. The FTP session remains connected and additional host commands will be processed.
16	Control port connection terminated. The session with the FTP server terminated. The data port connection was closed. Data transfer commands may not have completed. Check LASTRC and MAXRC to determine the state of transfer commands.
20 or greater	Immediate shutdown requested or a session termination error was detected locally. No further host commands will be processed. The current host command may not have completed. Check LASTRC and MAXRC to determine the success or failure of the command.

Replies received from the remote FTP server are documented in the FTP RFC 959. *Connect:Enterprise for z/OS Messages and Codes Guide* also contains information on remote FTP server replies.

The text replies generated by the local system are a three digit number followed by explanatory text. Replies that begin with a zero (0nn) indicate that the command completed successfully. Replies that begin with a six (6nn) indicate that the command failed.

Example with Sample Scripts and ODF

This example shows a sample LOGON_SCRIPT, AC_SCRIPT, and Options Definition File, followed by an explanation of the script processing. A sample dialog trace for this session is also shown.

Sample LOGON_SCRIPT

The following LOGON_SCRIPT, which is part of this example, is set up to open a connection with the remote FTP server and validate the user.

```
/*  REXX Sample – FTPLOGON  */
“userlog Running Logon Script” scriptname“.”
“open” ipaddr“,”portno
if HCRC > 0 then do
           “userlog OPEN for IP Addr/Port” ipaddr“/”portno “failed.”
           exit 4
end
“user” userid
if HCRC > 0 then do
           “userlog USER command for USERID=”userid “failed.”
           exit 4
end
“pass” password
if HCRC > 0 then do
            “userlog PASS command for USERID=”userid “failed.”
            exit 4
end
“acct 9945688”
exit 0
```

Sample AC_SCRIPT

The following AC_SCRIPT defines how and where information is to be transferred between the remote FTP server and Connect:Enterprise .

```
/* REXX SAMPLE SCRIPT – SENDONLY */

“TYPE E”
“MODE S”
“STRU R”
“LOCCD MBOXA”
“CD BREMOTE”
“PUT /MBOXA/*” FILE1
IF HCRC > 0 THEN DO
   “USERLOG PUT FAILED FOR MBOXA”
    EXIT 8
END
“QUIT”
EXIT 0
```

Sample ODF

The Options Definition File provides specific parameter values and generic script variables used to execute the Auto Connect session, such as the IP address and port number for the FTP remote server to connect to.

```
** Sample ODF *******
*OPTIONS
….
FTP_AC_SCRIPT_DEFAULT=SENDRECV
FTP_LOGON_SCRIPT_DEFAULT=FTPLOGON
….
….
*CONNECT
LISTNAME=FTPLIST1
TYPE=FTP
    FTPRMTB AC_SCRIPT=SENDONLY
*REMOTES
NAME=FTPRMTB
   TYPE=FTP_SERVER
   &IPADDR=MVSA
   &PORTNO=5566
   &USERID=BREMOTE
   &PASSWORD=RMTBPWD
```

The LOGON_SCRIPT and AC_SCRIPT perform the following steps using the script variables provided by the ODF when the Auto Connect FTPLIST1 is run.

1. A user log record is written indicating that the FTPLOGON LOGON_SCRIPT is running. FTPLOGON executes because it is set as the default LOGON_SCRIPT in the *OPTIONS section of the ODF, and the *REMOTES definition for remote FTPRMTB does not specify a LOGON_SCRIPT.
2. Because the remote definition for FTPRMTB in the ODF specifies &IPADDR=MVSA and &PORTNO=5566, the OPEN command in the FTPLOGON script results in a connection with the remote FTP server with IP address MVSA and port number 5566.

   If the OPEN command fails, a log record is created with the text “OPEN for IP Addr/Port MVSA/5566 failed.” and the FTP session ends.
3. Because the ODF remote definition for FTPRMTB specifies &USERID=BREMOTE, the USER command in the FTPLOGON script issues the “user BREMOTE” command to begin the logon process.

   If the user command fails, a log record is created with the text “USER command for USERID=BREMOTE failed.” and the FTP session ends.
4. Because the ODF remote definition for FTPRMTB specifies &PASSWORD=RMTBPWD, the PASS command in the FTPLOGON script results in the “PASS RMTBPWD” command being issued to validate the user. The password is reported as asterisks in the dialog trace (“PASS ********”).

   If the PASS command fails, a log record is created with the text “PASS command for USERID=BREMOTE failed.” and the FTP session ends.
5. Optional accounting information is supplied using the ACCT command.

6. If the FTPLOGON script completes with no errors, the SENDONLY AC_SCRIPT is run. SENDONLY is used instead of SENDRECV because the AC_SCRIPT is specified on the remote specification record.
7. The transfer type is set to EBCDIC.
8. The transfer mode is set to Stream.
9. The transfer structure is set to Record.
10. The local current working directory (Mailbox ID) is set to MBOXA.
11. The current working directory of the remote FTP server is changed to BREMOTE.
12. All requestable batches in mailbox MBOXA are sent to the remote FTP server. They are concatenated into a single file. The remote server stores the file as “file1” in the BREMOTE directory.
13. The FTP session ends.
14. The script ends.
15. The Auto Connect ends.

Excerpts from the dialog trace for this session are shown below.

Note: Actual release numbers are replaced by *xx.xx.00* (or *x.x.00*) in the excerpts below to designate the Connect:Enterprise for z/OS version you are using.

```
      13:41:44:49  COMMAND FROM SCRIPT: userlog Running Logon Script  FTPLOGON.
      13:41:44:54  COMMAND FROM SCRIPT: open MVSA, 5566
      13:41:44:59              DATE:  2001290
220-Connect:Enterprise  xx.xx.00 on OS/390 021000
220-Connection will close if idle for more than 0300 seconds.
220-Ready (local host date and time) 2001/290   at 13:41:45
220-YOU ARE LOGGED ONTO C:E FTP SERVER COMPANY B.
220                         SSL USE IS OFF.
     13:41:44:92  COMMAND FROM SCRIPT: user BREMOTE
     13:41:44:92    FTP CLIENT OUTPUT: user BREMOTE
     13:41:45:19              DATE:  2001290
331 Send password please.
     13:41:45:24  COMMAND FROM SCRIPT:  PASS *********************************
     13:41:45:24    FTP CLIENT OUTPUT:  PASS ********************************
     13:41:45:29              DATE:  2001290
230 BREMOTE  is logged on.  Current working Mailbox is “BREMOTE”.
     13:41:45:30    FTP CLIENT OUTPUT:  SITE IDENT PROD_ID=1 PROD_REL=x.x.00
     13:41:45:32              DATE:  2001290
200 PROD_ID=1 PROD_REL=1.1.00 PROC_LVL=1
     13:41:45:36  COMMAND FROM SCRIPT: acct 9945688
     13:41:45:37    FTP CLIENT OUTPUT: acct 9945688
     13:41:45:39              DATE:  2001290
502 ACCT command not supported.
      13:41:45:61  COMMAND FROM SCRIPT: type E
      13:41:45:61    FTP CLIENT OUTPUT: type E
      13:41:45:64              DATE:  2001290
200 Data representation type is E.
      13:41:45:69  COMMAND FROM SCRIPT: mode S
      13:41:45:69    FTP CLIENT OUTPUT: mode S
      13:41:45:71              DATE:  2001290
200 Data transfer mode is S.
      13:41:45:76  COMMAND FROM SCRIPT: stru R
      13:41:45:76    FTP CLIENT OUTPUT: stru R
      13:41:45:78              DATE:  2001290
200 Data structure is R.
      13:41:45:84  COMMAND FROM SCRIPT: loccd MBOXA
      13:41:45:85   050 Local Working MAILBOX_ID is MBOXA
      13:41:45:91  COMMAND FROM SCRIPT: cd BREMOTE
      13:41:45:91    FTP CLIENT OUTPUT: CWD BREMOTE
      13:41:45:92              DATE:  2001290
250 CWD was successful. Current working Mailbox is “BREMOTE”.
      13:41:45:97  COMMAND FROM SCRIPT: put *
      13:41:46:00    FTP CLIENT OUTPUT: SITE LRECL=00080 BLKSIZE=06320 RECFM=F
      13:41:46:02              DATE:  2001290
200 SITE command was accepted.
      13:41:46:05    FTP CLIENT OUTPUT: PORT 10,20,200,2,32,84
      13:41:46:07              DATE:  2001290
200 PORT request OK (10,20,200,2,32,84).
      13:41:46:07    FTP CLIENT OUTPUT: STOR ’JULY INVOICE           ’.
      13:41:46:14              DATE:  2001290
150 Opening data connection.  Storing ’JULY INVOICE’ as batch number 0000015.
      13:41:46:16  FTP DATA SENT:
0B378000  D9C5C3D6 D9C440F0 F140D6C6 40F0F440   40404040 40404040 40404040 40404040
.
.
.
      13:41:46:17  FTP DATA SENT:
0B378000  FF02
      13:41:46:24  FTP DATA SENT:
0B378000  D9C5C3D6 D9C440F0 F140D6C6 40F0F440   40404040 40404040 40404040 40404040
.
.
.
0B3780E0  40404040 40404040 40404040 40404040   40404040 FF01D9C5 C3D6D9C4 40F04040
0B378100  D6C640F0 F4404040 40404040 40404040   40404040 40404040 40404040 40404040
0B378120  40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040
0B378140  40404040 4040FF01
0B378000  FF02
 226 Transfer complete.  ’JULY
 INVOICE’, batch number 0000015  660 bytes.
      13:41:46:41  COMMAND FROM SCRIPT: quit
      13:41:46:41    FTP CLIENT OUTPUT: quit
      13:41:46:45              DATE:  2001290
 221 QUIT command received. Goodbye.
      13:41:46:53  CLOSING TRACE
```

Sample Scripts in the EXAMPLE Library

The following sample scripts are provided in the EXAMPLE library for creating and customizing logon and session scripts:

Example Member Name	Script Type	Use
PASSTABL	LOGON_SCRIPT	Illustrates possible ways to maintain remote passwords in a table without using &PASSWORD or &NEWPASS variables.
DELAY	LOGON_SCRIPT	Illustrates how to delay retry when an Auto Connect session fails.
FTPLOGON	LOGON_SCRIPT	Logs on to a remote FTP server using symbolics set in the remote definition and illustrates how to use the &PASSWORD, &NEWPASS, and &USERID variables.
FTPACVAR	AC_SCRIPT	Shows the values passed in each Connect:Enterprise REXX variable.
HOSTADDR	LOGON	Illustrates the use of built-in REXX socket calls to obtain the IP address of a host name. The default returns the system's IP address.
HOSTNAME	LOGON	Illustrates the use of built-in REXX socket calls to obtain the host name of an IP address. The default returns the system's host name.
NOBATCH	LOGON_SCRIPT	Simulates NOBATCH=NC. Performs a local DIR and if no batches are available to send, does not sign on.
RETRY	LOGON_SCRIPT	Illustrates how to retry when an Auto Connect session fails.
IEFBR14	LOGON_SCRIPT	Blank script to use as the LOGON_SCRIPT if you put your logon commands in the AC_SCRIPT.
RECVONLY	AC_SCRIPT	Establishes session parameters and receives data from the remote FTP server.
SENDRECV	AC_SCRIPT	Establishes session parameters and receives data from the remote FTP server, then sends data to the remote FTP server.
LIST1	AC_SCRIPT	Illustrates how to use &IDLIST only.
LIST2	AC_SCRIPT	Illustrates how to use &BEGINLST and &IDLIST. &BEGINLST is only sent when IDLIST batches are available.
LIST3	AC_SCRIPT	Illustrates how to use &BEGINLST, &IDLIST, and &ENDLIST. &BEGINLST/&ENDLIST is only sent when IDLIST batches are available.
LIST4	AC_SCRIPT	Illustrates how to use just &IDLIST and &ENDLIST. &ENDLIST is only sent when &IDLIST batches are available.
LOCDIR	AC_SCRIPT	Illustrates how to use LOCDIR to PUT only those batches which have not already been transmitted.

Example Member Name	Script Type	Use
SENDONLY	AC_SCRIPT	Establishes session parameters and sends data to the remote FTP server.
FIREWALL	LOGON_SCRIPT	Logs on to a remote FTP server going through a proxy firewall.
UNIXFTP	AC_SCRIPT	Illustrates how to send or receive data from a UNIX FTP server.
UNIXRDX	AC_SCRIPT	Illustrates how to send or receive data from a Connect:Enterprise UNIX FTP server.

Script Command Details

This section lists the commands that can be used in a LOGON_SCRIPT and an AC_SCRIPT. For each script command, its format including abbreviations, a description, and examples are given. Additional information, such as selection criteria, is provided for some commands.

Many of these commands use the </MID/BID> | <BID> parameter. The </MID/BID> | <BID> parameter consists of a 1–8 character Mailbox ID enclosed in slashes (/MID/) followed by a 1–64 character Batch ID (BID) or the Batch ID by itself.

The Mailbox ID overrides the current working directory on the local system for the duration of the command.

The Batch ID selects batches for the command. The Batch Selection heading in the following commands describes how the Batch ID is used to select batches.

The Batch ID is case-sensitive. Enclose the Batch ID in quotes if it contains a dash (-), period (.), asterisk (*), question mark (?), open bracket ([), close bracket (]), forward slash (/), or backward slash (\), or uses a pound sign (#) as its first character.

Specify the Batch ID in one of the following ways:

Format	Description
nnnnnnn	The 1–7 digit batch number from 1–9999999. Leading zeros are optional.
“generic-name”	The 1–64 characters used to identify batches in the current working mailbox. The quotation marks are required for this form of Batch ID. Do not use wildcard characters in the “generic-name” value.
‘specific’	The 1–64 character Batch ID left justified, padded with trailing blanks. Single quotes are required to delimit the Batch ID. If the ID contains an apostrophe, code two apostrophes for each single apostrophe in the Batch ID. For example, code UID1235’BNK1 as ‘UID1235”BNK1’.

Format	Description
‘SPEC*?[ ]’	Indicates any batch that matches the specified pattern. Consists of 1–24 characters including any wildcard card characters. The pattern is compared to all batches in the current working Mailbox ID and any matching batches are selected.
‘batch_ID.#nnnnnnn’	The 1–64 character Batch ID with trailing blanks truncated, followed by .# and the 7-digit batch number from 0000001–9999999 (including leading zeros) and enclosed in single quotes. You must use single quotes for embedded blanks in this Batch ID.
‘#nnnnnnn.dat’	A pound sign (#) and the 7-digit batch number from 0000001–9999999 with leading zeros followed by .dat and optionally enclosed in single quotes.

ACCT

Format: ACCT <account info>

where <account info> is any account information required by the remote FTP server.

Description: Provides the necessary TELNET string identifying the user’s account to a server. Most connections do not require it. The remote server should indicate the need for this command and what account information is required before you include it in a script.

Examples: “ACCT AC124354”

ASCII

Format: ASCII

ASC

AS

Description: Changes the current data transfer type to ASCII.

All ASCII batches received from a remote FTP server are translated to EBCDIC before being added to the local working mailbox.

Character batches are translated to ASCII before being sent to a remote FTP server.

Examples: “ASCII”

BINARY

Format: BINARY

BIN

B

Description: Changes the current data transfer type to image.

All batches received from a remote FTP server are added to the current working mailbox without translation. Batches are sent to the remote FTP server without translation.

Character batches are sent to the remote FTP server without translation (as EBCDIC).

Examples: “BINARY”

CD

Format: CWD <directory>

CW <directory>

where <directory> specifies the name of a file directory at the remote FTP server.

Description: Changes the current working directory at the remote FTP server. File transfers are done to and from the current working directory.

The remote FTP server defines the <directory> syntax requirements. When communicating with a Connect:Enterprise FTP server the <directory> parameter is the Mailbox ID on the remote.

Examples: “CD MBOXA” Changes a remote FTP server directory to MBOXA.

“CD” BEGINLIST Changes a remote FTP server directory to the value of the ODF &BEGINLIST variable.

CDUP

Format: CDUP

Description: Changes the remote FTP server’s current working directory to the next higher directory level. The Connect:Enterprise FTP server does not support the CDUP command.

Example : “CDUP”

DELETE

Format: DELETE <foreignfile>

DELE <foreignfile>

where <foreignfile> specifies the name of the file to be deleted from the remote FTP server.

Description: Deletes a file on the remote FTP server.

The remote FTP server defines the syntax requirements for the <foreignfile> parameter. If the remote FTP server is Connect:Enterprise :

- To delete a file from the current working mailbox, specify <BID> for <foreignfile>.
- To delete a file from a mailbox other than the current working mailbox, specify </MID/BID> for <foreignfile>.

Examples: “DELETE /MBOXA/List*”—deletes all files that begin with “List” from mailbox MBOXA.

“DELE List*”—deletes all files that begin with “List” from the current working directory.

“DELETE test.june12.000001”—deletes the test.june12.000001 file from the remote FTP server.

DIR

Format: DIR [<directory>]

DI [<directory>]

where <directory> specifies the name of the directory or files on the remote FTP server to be listed.

Description: Sends the server LIST command to obtain a list of the directory entries or files from the remote FTP server.

When this command is executed, a batch is added to the currently working Mailbox ID with a Batch ID of “LIST + first 19 bytes of <directory>”. The batch contains the directory listing or file list from the remote FTP server. It also places the directory listing contents into REXX compound variables with the stem ‘DIR.’ Each compound variable has a single level numeric extension and contains one content line. The content variables are DIR.1 through DIR.*n*. DIR.0 contains the number of content lines. For more information on the DIR variables, see *REXX LOCDIR. and DIR.Variables* on page 219 and *Additional DIR. Variable Information* on page 220.

The remote FTP server defines the syntax requirements for the <directory> parameter. If the remote FTP server is Connect:Enterprise :

- To list batches from the remote current working mailbox, omit the <directory> parameter. The added batch’s Batch ID is just “LIST”.
- To list specific Batch IDs from the remote current working mailbox, specify <BID> for <directory>.
- To list all batches from a mailbox other than the current working mailbox, specify </MID/> for <directory>.
- To list specific Batch IDs from a mailbox other than the current working mailbox, specify
  </MID/BID> for <directory>.

By default, the DIR command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: “DIR”—lists batches from the remote current working directory.

“DIR MBOXA”—lists batches from the MBOXA remote directory.

“DIR ‘ABC Company Invoice’ ”—lists files named ABC Company Invoice from the remote current working directory.

EBCDIC

Format: EBCDIC

EBC

EB

Description: Changes the current data transfer type to EBCDIC.

All batches received from a remote FTP server are added to the current working mailbox without translation.

Character batches are not translated before being sent to the remote FTP server.

Binary formats are not sent to the remote FTP server when the data transfer type is set to EBCDIC.

Examples: “EBCDIC”

GET

Format: GET <foreignfile> </MID/BID> | <BID>

G <foreignfile> </MID/BID> | <BID>

where <foreignfile> specifies the name of the file to be retrieved from the remote FTP server.

</MID/BID> specifies the Mailbox ID and the Batch ID of the batch to be added.

<BID> specifies the Batch ID of the batch to add to the current working mailbox.

Both the <foreignfile> and the </MID/BID> | <BID> parameters are required.

Description: Collects a file from a remote FTP server and adds it under the specified Batch ID to a mailbox .

The remote FTP server defines the syntax requirements of the <foreignfile> parameter. If the remote FTP server is Connect:Enterprise :

- Specify </MID/BID> to collect a file from a mailbox other than the remote’s current working directory.
- Specify <BID> when collecting from the remote server’s current working mailbox

Any data specified after the second parameter is ignored.

Several LOCSITE parameters affect GET processing and file characteristics. For more information, see page 238.

By default, the GET command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: “GET /ABC/BillData /MBOXB/PhoneBill”—gets data from the ABC mailbox with a Batch ID of BillData from the remote Connect:Enterprise FTP server and adds it to the MBOXB mailbox with a Batch ID of PhoneBill on the local Connect:Enterprise client.

“GET InvoiceData /SALES/AprilInvoice”—gets the InvoiceData file from a remote FTP server and adds it to the SALES mailbox with a Batch ID of AprilInvoice.

“GET InvoiceData SalesInvoice”—gets the InvoiceData file from remote FTP server and adds it to current working mailbox with a Batch ID of SalesInvoice.

“GET InvoiceData /SALES/’March Invoice’ ”—gets InvoiceData file from remote FTP server and adds it to the SALES mailbox with a Batch ID ‘March Invoice’.

IMAGE

Format: IMAGE

Description: Changes the current data transfer type to image.

All batches received from a remote FTP server are added to the current working mailbox without translation as data type image.

All batches are sent to the remote FTP server without translation.

Examples: “IMAGE”

LIST

Format: LIST [<directory>]

LI [<directory>]

where <directory> specifies the name of the directory or files on the remote FTP server to list.

Description: Obtains a list of the directory entries or files on the remote FTP server. This command adds a batch to the currently working Mailbox ID with a Batch ID of “LIST [+ first 19 bytes of <directory>].” The batch will contain the directory listing or list of files from the remote FTP server.

The remote FTP server defines the syntax requirements for the <directory> parameter. If the remote FTP server is Connect:Enterprise :

- To list batches from the remote current working mailbox, omit the <directory> parameter. The added batch’s Batch ID is just LIST
- To list specific Batch IDs from the remote current working mailbox, specify <BID> for <directory>.
- To list all batches from a mailbox other than the current working mailbox, specify </MID/> for <directory>.
- To list specific Batch IDs from a mailbox other than the current working mailbox, specify </MID/BID> for <directory>.

By default, the LIST command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server

PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: “LIST Test*”—lists all files that start with Test on the remote server’s current working directory.

LOCCD

Format: LOCCD <MID>

where <MID> specifies the 1–8-character Mailbox ID.

Description: Changes the local current working mailbox to the value specified in <MID>. Subsequent commands use the specified Mailbox ID as the current working mailbox. You do not need to type slashes before or after the Mailbox ID. The Mailbox ID does not have to exist.

Two periods in a row (..) or one period between two blank characters (.) are not supported and are treated as a syntax errors. Do not use the hyphen (-), period (.), asterisk (*), question (?), left bracket ([), right bracket (]), forward slash (/) or backward slash (\) in the Mailbox ID.

Examples: “LOCCD INVOICE”—changes the local current working directory to INVOICE.

LOCDIR

Format: LOCDIR [</MID/BID> | <BID>]

where </MID/BID> specifies the Batch IDs to be listed from the specified Mailbox ID

<BID> specifies the Batch IDs to be listed from the local current working mailbox.

Description: Lists the contents of the local current working mailbox or the mailbox specified in the </MID/BID> parameter. It also places the contents into REXX compound variables with the stem ‘LOCDIR.’ Each compound variable has a single level numeric extension and contains one content line. The content variables are LOCDIR.1 through LOCDIR.*n*. LOCDIR.0 contains the number of content lines. For more information on the LOCDIR variables, see *REXX LOCDIR. and DIR.Variables* on page 219. You can set the local current working mailbox with the LOCCD command. The mailbox specified in </MID/BID> overrides the local current working mailbox for this command, but does not permanently change the local current working mailbox.

The AC_SCRIPT LOCDIR command performs functions similar to the FTP server LIST command. Like the FTP server LIST processing, the values for DIR_FILTER, FTIME, ORIGIN, and TTIME are maintained for the session used to populate the list. The local parameters have default values that can be changed with the LOCSITE command and displayed using the LOCSTAT command.

Different List Formats Returned by LOCDIR

The format of the list returned by the LOCDIR command depends on the value of the DIRFORM parameter of the LOCSITE command. The format returned when

DIRFORM=MBOX_ZOS is shown in the following example, which illustrates the default directory list format.

```
  (1)      (2)        (3)        (4)           (5)       (6)          (7)     (8)
COMPANYB #0000047 CT=00000320 BID=LIST TEST  1508-01199  A R    F  O38  VBQ01 OFFLINE
COMPANYB #0000067 CT=00000420 BID=LIST       1046-01205  C  T   F  O38  VBQ20
COMPANYB #0000068 CT=00000210 BID=LIST       1046-01205  C      F  O38  VBQ03
```

Column	Description
(1)	Mailbox where the batch resides.
(2)	Batch number–zero filled.
(3)	Size of batch in bytes.
(4)	Batch ID
(5)	Date batch added (HHMM-YYDDD)
(6)	Batch flags in 20 character field in the following order: I = Incomplete A|C = Offline Added, Online Collected D = Delete N = Not Transmittable U = Unextractable R = Requestable T = Transmitted E = Extracted M = Multi-transmittable X = Transparent B|F|S|Z = BSC, FTP, SNA or API Space = Reserved for future use. Space = Reserved for future use. 0 = File oriented data. 1|2|3 =FTP mode – blocked, compressed, stream 8|9 = FTP structure – file, record e = Offline added with encryption (ENCR=) Space = Reserved for future use. Space = Reserved for future use. Space = Reserved for future use.
(7)	The number of the VSAM Batch Queue where the batch is stored.
(8)	VBQ status–OFFLINE if VBQ is not available or blank if available.

The following example illustrates the format returned when DIRFORM=BROWSER; the table provides an explanation of the columns.

```
(1)       (2)  (3)     (4)     (5)        (6)       (7)
A R       1 COMPANYB 0000047 00000320 JUL 18 15:08 LIST TEST
C  T      1 COMPANYB 0000067 00000420 JUL 24 10:46 LIST
IC        1 COMPANYB 0000074 00000320 JUL 24 11:31 July Invoice
```

Column	Description

(1)	Batch flags in 10-character field in the following format:
	I = Incomplete
	A|C = Offline added, Online Collected
	D = Delete
	R = Requestable
	T = Transmitted
	E = Extracted
	M = Multi-transmittable
	U = Unextractable
	N = Not transmittable
	X = Transparent

(2)	Constant 1.
(3)	Mailbox where batch resides.
(4)	Batch number–zero filled.
(5)	Size of batch in bytes.
(6)	Date batch added (MMM DD HH-MM)
(7)	Batch ID

The format returned when DIRFORM=MBOX_CLIENT is shown below with an explanation of the columns.

```
(1)         (2)        (3)        (4)               (5)       (6)      (7)
COMPANYB #0000047 CT=00000320 BID=LIST TEST      1508-01199  A R   COMPANYB
COMPANYB #0000067 CT=00000420 BID=LIST           1046-01205  C  T  COMPANYB
COMPANYB #0000068 CT=00000210 BID=LIST           1046-01205  C     COMPANYB
```

Column	Description
(1)	Mailbox where batch resides.
(2)	Batch number–zero filled.

Column	Description
(3)	Size of batch in bytes.
(4)	Batch ID
(5)	Date batch added (HHMM-YYDDD)

(6)	Batch flags in 10-character field in the following order:
	I = Incomplete
	A|C = Offline Added, Online Collected
	D = Delete
	R = Requestable
	T = Transmitted
	E = Extracted
	M = Multi-transmittable
	U = Unextractable
	N = Not transmittable
	X = Transparent

(7)	Batch job name or remote name that added the batch.

Batch Selection: If the specified Batch ID is in the format ’#nnnnnnn’, ’nnnnnnn’, ’BID.#nnnnnnn’, or ’#nnnnnnn.dat’ and the batch is in the current working mailbox, the specified batch is eligible to be listed.

If the specified Batch ID is in the format “generic-name”, ’specific’, or ’spec*?[]’, all batches in the current working mailbox that match the Batch ID are eligible to be listed.

If the Batch ID is not specified, all batches in the current working mailbox are eligible to be listed.

Eligible batches are selected depending on the settings of the DIR_FILTER, FTIME, ORIGIN, and TTIME parameters of the LOCSITE command. An eligible batch is listed if the following conditions are met:

- The batch’s status flags do not match any of the values specified in the DIR_FILTER parameter.
- The batch’s creation date and time is on or after the value specified in the FTIME parameter.
- The batch’s creation date and time is on or before the value specified in the TTIME parameter.
- The batch’s origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.

If the Batch ID is in the format ’#nnnnnnn’, ’nnnnnnn’, ’BID.#nnnnnnn’, or ’#nnnnnnn.dat’, the DIR_FILTER, FTIME, ORIGIN, and TTIME parameters are ignored.

If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive completion reply when all of the selected batches have been listed.

Examples: “LOCDIR”—lists all contents of the current working mailbox.

“LOCDIR /MBOXA/#0000023”—lists batch 23 in mailbox MBOXA.

“LOCDIR 0000043.dat”—lists all files named 0000043.dat.

LOCPWD

Format: LOCPWD

Description: Displays the local current working Mailbox ID.

Examples: “LOCPWD”

LOCSITE

Format: LOCSITE [<parameter> <parameter> … …]

LOCSI [<parameter> <parameter> … …]

where [<parameter> <parameter>… …] specifies one or more Connect:Enterprise site-specific parameters

Description: Sets Connect:Enterprise -specific parameters that affect the following AC_SCRIPT commands:

- The GET and MGET commands that add batches (collect)
- The MPUT and PUT commands that send batches (transmit)
- The LOCDIR command that lists batches

Specifying the LOCSITE command with the RESET parameter sets all of the LOCSITE parameter values to the session default value. The default for each of the parameters is determined by the values specified in the remote’s definition in the *REMOTES section of the ODF.

Use the LOCSTAT command to view the current LOCSITE command parameters values. Specifying the LOCSITE command with no parameters gives the same results as the LOCSTAT command.

The values set for a LOCSITE command parameter remain in effect for the remainder of the session, except for the DIR_FILTER parameter. The DIR_FILTER changes are only effective for the next execution of the LOCDIR and MPUT command, respectively. To make the DIR_FILTER value effective for the remainder of the session, specify the KEEP option.

The following table lists the AC_SCRIPT commands column affected by each LOCSITE parameter.

LOCSITE Parameter	Description	AC_SCRIPT Command
BCHSEP=NONE | OPT3 | OPT4	Specifies batch separation options for batches being transmitted.	MPUT, PUT
BLKSIZE=0-32760 | blank	Specifies block size of file being received.	GET, MGET
BLOCKS	Specifies primary allocation amount of file being received is in BLOCKS.	GET, MGET
BLOCKSIZE=0-32760 | blank	Specifies block size of file being received.	GET, MGET
CYLINDERS	Specifies primary allocation amount of file being received is in CYLINDERS.	GET, MGET
DIR_FILTER=D | <flags> [KEEP]	Specifies batch exclusion filter.	LOCDIR
DIR_FILTER DEFAULT | NONE | OFF [KEEP]	Resets batch exclusion filter.	LOCDIR
DIR_FILTER [QUERY]	Displays batch exclusion filter	N/A
DIRECTORY=1-16777215 | blank	Specifies number of directory blocks allocated for file being received.	GET, MGET
DIRFORM=BROWSER | BROWSER64 | MBOX_CLIENT | MBOX_CLIENT64 | MBOX_ZOS | MBOX_ZOS64 | $MIBNSDFXY | UNIX | UNIX64	Specifies the directory format.	LOCDIR
EDI=NO | YES	Specifies whether single byte x'15' segment terminators are used.	MPUT, PUT
EO=NO | YES	Specifies extract once option for batches being received.	GET, MGET, SCGET
FTIME=blank | <from_time>	Specifies batch selection criteria for transmission.	LOCDIR, MPUT, PUT, SCPUT
KALIVEOFF	Turns off the KALIVEON parameter. **Note:** KALIVEOFF is equivalent to setting KALIVEON=0.	LOCSITE

LOCSITE Parameter	Description	AC_SCRIPT Command
KALIVEON=nnnnnnn	Keeps remote control ports open for the specified number of seconds (0–2147460) for non-Connect:Enterprise applications that experience timeouts on their control ports even though they have data port activity.	LOCSITE
KIRN=YES | NO	Specifies if Connect:Enterprise keeps or removes record separator strings in batches.	GET, MGET
LRECL=0-32760 | blank | “x”	Specifies the LRECL of file being received.	GET, MGET
MULTXMIT=NO | YES	Specifies the multiple transmission flag for batches being received.	GET, MGET, SCGET
ONEBATCH=NO | YES	Specifies if multiple batches should be transmitted.	MPUT, PUT, SCPUT
ORIGIN=<originator> | blank	Specifies the batch selection criteria for transmission.	LOCDIR, MPUT, PUT, SCPUT
PRIMARY=1-16777215 | blank	Specifies the primary allocation amount of file being received.	GET, MGET
RECFM=<recfm> | blank	Specifies the RECFM value for file being received.	GET, MGET
RESET	Set all values to their default.	LOCDIR, GET, MGET, MPUT, PUT, SCGET, SCPUT
REMOTE_FILENAME_LENGTH= SHORT | LONG | LONG64	Specifies the foreign file name’s length.	MPUT, PUT
RIFS=YES | NO	Changes the batch to record structure or retains the batch as file structure.	GET, MGET
SCAN=NO|YES|ALL	Specifies whether the Connect:Enterprise FTP client scans RETR received batches for $$ commands and /* cards.	GET, MGET
SECONDARY=1-16777215 | blank	Specifies the secondary allocation amount of the file being received.	GET, MGET
TO=NO |YES	Specifies the transmission control limits for the batch being received.	GET, MGET, SCGET

LOCSITE Parameter	Description	AC_SCRIPT Command
TTIME=<to_time> | blank	Specifies the batch selection criteria for batches being transmitted.	LOCDIR, MPUT, PUT, SCPUT
TRACKS	Specifies the primary allocation amount in TRACKS of the file being received.	GET, MGET
VBQ#=nn	Specifies the VBQ on which batches collected in FTP sessions are to be stored.	DIR, GET, LIST, NLST, MGET, SCGET
XMIT=NO | YES	Specifies the transmission control limits for the batch being received.	GET, MGET, SCGET

The following table lists AC_SCRIPT commands and what LOCSITE parameters affect them.

AC_SCRIPT command	is affected by LOCSITE parameter

LOCDIR	DIR_FILTER=D | <flags> [KEEP]
	DIR_FILTER DEFAULT | NONE | OFF [KEEP]
	DIRFORM=BROWSER | MBOX_CLIENT | M BOX_ZOS | $MIBNSDF
	FTIME=<from_time>
	ORIGIN=<originator>
	RESET
	TTIME=<to_time>

AC_SCRIPT command	is affected by LOCSITE parameter

GET
MGET
	BLKSIZE= 0-32760 | blank
	BLOCKS
	BLOCKSIZE= 0-32760 | blank
	CYLINDERS
	DIRECTORY= 1-16777215 | blank
	EO= NO | YES
	KIRN=YES | NO
	LRECL= 0-32760 | blank | “x”
	MULTXMIT= NO | YES
	PRIMARY= 1-16777215 | blank
	RECFM=<recfm_list> | blank
	RESET
	RIFS=YES | NO
	SCAN=NO|YES|ALL
	SECONDARY= 1-16777215 | blank
	TO= NO | YES
	TRACKS
	VBQ#=nn
	XMIT= NO | YES

PUT
MPUT
	BCHSEP=NONE | OPT3 | OPT4
	FTIME=<from_time>
	DIR_FILTER=D | <flags> [KEEP]
	DIR_FILTER DEFAULT | OFF | NONE [KEEP]
	ONEBATCH=NO | YES
	ORIGIN=<originator>
	REMOTE_FILENAME_LENGTH=LONG | SHORT
	RESET
	TTIME=<to_time>

SCPUT	DIR_FILTER=D | <flags> [KEEP]
	DIR_FILTER DEFAULT | OFF | NONE [KEEP]
	FTIME=<from_time>
	ORIGIN=<originator>
	RESET
	TTIME=<to_time>

The following table categorizes the LOCSITE parameters according to whether they are related to sending or receiving files or the file characteristics of batches.

LOCSITE Parameters Related to Receiving Files	LOCSITE Parameters Related to Sending Files	LOCSITE Parameters Related to File Characteristics of Batches
EO	BCHSEP	BLKSIZE
KIRN	FTIME	BLOCKS

LOCSITE Parameters Related to Receiving Files	LOCSITE Parameters Related to Sending Files	LOCSITE Parameters Related to File Characteristics of Batches
MULTXMIT	KIRN	BLOCKSIZE
TO	ONEBATCH	CYLINDERS
RIFS	ORIGIN	DIRECTORY
SCAN	REMOTE_FILENAME_LENGTH	LRECL
VBQ#	RIFS	PRIMARY
XMIT	SCAN	RECFM
	TTIME	SECONDARY
	XMIT	TRACKS

The following table describes the LOCSITE command parameters in detail:

Parameter	Description
BCHSEP=NONE | OPT3 | OPT4	BCHSEP=NONE specifies that for a single connection, the PUT or MPUT command concatenates multiple batches to form a single batch. Each batch is flagged as transmitted when it is sent successfully. If a failure occurs during transmission, BCHSEP=NONE prevents batches previously sent from being retransmitted. If the transmission is restarted to the same remote client file, the data from the previous batches is replaced with the data from the unsent batches, and the data from the previously sent batches is lost. If the transmission is restarted to a different remote client file, data sent in previous batches will not be lost. BCHSEP=OPT3 specifies that for a single connection, the PUT or MPUT command concatenates multiple batches to form a single batch. All batches are flagged as transmitted when all batches have been sent successfully. If a failure occurs during transmission, BCHSEP=OPT3 will allow all batches to be transmitted again. If the transmission is restarted to the same remote client file, the data from the previous batches is retransmitted and no data will be lost. If the transmission is restarted to a different remote client file, duplicate records will be sent to the remote site. BCHSEP=OPT4 specifies that all batches be sent as individual files. It also affects batches selected by the PUT and MPUT commands and the format of the foreign file name sent to the remote FTP server. The remote definition and the Auto Connect definition determine the default for this parameter.

Parameter	Description
BLKSIZE=0–32760 | blank	Specifies the block size of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. A value of 0 indicates that Connect:Enterprise should determine an optimal block size for the file. Omit the equal sign if you specify BLKSIZE with a blank value. There is no default value for this parameter.
BLOCKS	Specifies that the primary and secondary allocation amounts of the batch being received are in BLOCKS. When Connect:Enterprise receives this information, it will store it with the batch information. No other processing is done. There is no default value for this parameter.
BLOCKSIZE=0–32760 or blank	This parameter is identical to BLKSIZE. See BLKSIZE for information on this parameter. There is no default value for this parameter.
CYLINDERS	Specifies that the primary and secondary allocation amounts for the batch being received are in CYLINDERS. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. There is no default value for this parameter.
DIR_FILTER=D | <flags> [KEEP]	This parameter is used by the LOCDIR command as a filter to exclude batches from the list. After the next LOCDIR command executes, the value of the DIR_FILTER parameter is set to the default value unless you specify the ’KEEP’ parameter. When you specify KEEP, the new flag settings remain in effect for the duration of the current session or until another LOCSITE DIR_FILTER command is encountered. This value is also used to select batches for the PUT, MPUT, and SCPUT commands. The default value for this parameter is D. **Note:** No batch is eligible for transmission using FTP if the STATFLAGs include I, D, N, or T. The batch must be flagged R. LOCSITE DIR_FILTER=R permits LOCDIR display of non transmittable batches. DIR_FILTER=R used in a PUT command results in no batches found.

Parameter	Description
DIR_FILTER DEFAULT | OFF | NONE [KEEP]	Changes the DIR_FILTER value to its default value. The parameters DIR_FILTER DEFAULT and DIR_FILTER DEFAULT KEEP are the same. The DIR_FILTER OFF parameter changes the value of the DIR_FILTER to nulls. This causes the next LOCDIR command to list all batches, regardless of the batch's status flag. The DIR_FILTER NONE sets the value of the exclusion flags to D. This causes the next LOCDIR command to list all batches except the batches with a status of delete. After the next LOCDIR command is executed, the value of the DIR_FILTER parameter is set to the default value unless the KEEP parameter is also specified. When the KEEP parameter is specified, the new flag settings take effect for the execution of the next LOCDIR command and remain in effect for the duration of the current session or until another LOCSITE DIR_FILTER command is encountered.
DIR_FILTER [QUERY]	The DIR_FILTER or DIR_FILTER QUERY parameters of the LOCSITE command list the current value of the DIR_FILTER flags.
DIRECTORY = 1–32760 or blank	Specifies the directory size of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. Omit the equal sign if you specify DIRECTORY with a blank value. There is no default value for this parameter.
DIRFORM = BROWSER | MBOX_CLIENT | MBOX_ZOS | $MIBNSDF	Specifies the format of a line written by the LOCDIR command. For information on the format and the fields returned for each DIRFORM value, see page 234. DIRFORM=BROWSER specifies that the LOCDIR command return the batch list in the format supported by browsers. DIRFORM=MBOX_CLIENT specifies that the LOCDIR command is to return the batch list in the format supported by Connect:Enterprise Client for Windows and the HTTP option. DIRFORM=MBOX_ZOS specifies that the LOCDIR command return the batch list in the Connect:Enterprise $$DIR format. This is the default value. DIRFORM=$MIBNSDF represents the variable directory format supported by the Connect:Enterprise FTP Server.
EO=NO | YES	EO=YES specifies that the batch being received can be extracted once and cannot be transmitted. The batch is marked non-transmittable when it is added and non-extractable after it has been extracted at the host site. The default value EO=NO does not prevent transmission or multiple extractions.

Parameter	Description
FTIME=<from_time>	Specifies the earliest creation date and time that a batch can have to be eligible for processing. Valid formats are: nnn—date calculated as nnn days prior to the current date. hhmm—time expressed in universal (also known as military) time format. For example, 11.15 PM is 22.15. When no date is specified, the date is 1980/001. yyddd—2-digit year and 3-digit Julian day. ccyyddd—2-digit century indicator (19 or 20), 2-digit year, and 3-digit Julian day. nnn:hhmm—number of days prior to current date and time expressed in universal time format. yyddd:hhmm—2-digit year, 3-digit Julian day, and time expressed in universal time format. ccyyddd:hhmm—2-digit century indicator (19 or 20), 2-digit year, 3-digit Julian day, and time expressed in universal time format. There is no default value for this parameter. **Note:** You cannot use two FTIME parameters to set time, for example, FTIME=1100 and FTIME=0134. These two separate parameters taken together do not indicate 11:00 on the 341st day of 2001. To indicate that time, you would use the *yyddd:hhmm* format to specify FTIME=01341:1100.
KIRN=YES | NO	Specifies if Connect:Enterprise keeps or removes record separator strings in batches. The default comes from the KIRN setting in the *REMOTES record. NO = Connect:Enterprise removes the record separator string after recordizing the batch so that the batch is stored as a file structure. YES = Connect:Enterprise keeps the Record separator strings in the batch and changes the batch to record structure. The corresponding RIFS parameter must be set to YES. See *Recordizing Batches* on page 113 for more information.
LRECL=0–32760 | blank | “x”	Specifies the logical record length of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. Omit the equal sign if you specify LRECL with a blank value. The “x” specification is ignored. There is no default value for this parameter.
MULTXMIT=NO | YES	MULTXMIT=NO specifies that the batch being received cannot be transmitted multiple times. MULTXMIT=YES specifies that a batch being received can be transmitted multiple times to one or more remote sites. The default value is MULTXMIT=NO.

Parameter	Description
ONEBATCH = NO | YES	ONEBATCH=YES specifies that only the first eligible batch can be selected for transfer to the remote FTP client. ONEBATCH=NO specifies that all eligible batches can be selected for transfer to the remote FTP client. The default value of this parameter is specified in the Auto Connect definition.
ORIGIN = <originator>	Specifies that only batches created by the specified originator are eligible for processing. There is no default value for this parameter. Batches from any origin will be eligible.
PRIMARY = 1–16777215	Specifies the primary allocation amount of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. There is no default value for this parameter.
RECFM = F | F A | FB | FBA | FBM | FBS | FBSA | FBSM | FM | FS | FSA | FSM | U | UA | UM | V | VA | VB | VBA | VBM | VBSA | VBSM | VBS | VM | VS | VSA | VSM	Specifies the record format of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. There is no default value for this parameter.
RESET	Specifies that all LOCSITE values be reset to their defaults. If no default exists for a given value, it is cleared to nulls.
REMOTE_FILENAME_ LENGTH = SHORT | LONG | LONG64	Specifies the format of the foreign file name created for batches transmitted by the MPUT and PUT commands when BCHSEP=OPT4 is specified. The foreign file name format when BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=LONG is <foreignfile.#nnnnnnn>, where <foreignfile> is the second parameter supplied on the PUT or STOR command, # is a constant, and nnnnnnn is the 7-digit batch number with leading zeroes. If <foreignfile> is not specified, it defaults to the 1–64 character Batch ID of the selected batch. The format of the list returned when BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=SHORT is specified is <#nnnnnnn.dat>, where “#“ is a constant, “nnnnnnn“ is the 7-digit batch number with leading zeroes, and “.dat“ is a constant. The default for this parameter is specified in the remote definition.

Parameter	Description
RIFS=YES | NO	(RIFS stands for Recordize Input File Structure) Changes the batch to record structure or retains the batch as file structure. The default comes from the RIFS setting in the *REMOTES record. YES = Recordizes the batch after recognizing a record separator string and uses CRLF for SFA batches and NL for SFE batches. NO = Retains file structure of batch and does not recognize record separator strings in SFA or SFE batches. See *Recordizing Batches* on page 113 for more information. **Note:** Processing results cannot be predicted or supported when RIFS=NO and SCAN is set to YES or ALL.
SCAN=NO|YES|ALL	Specifies whether the Connect:Enterprise FTP client scans RETR received batches for $$ commands and /* cards. NO = Scanning for Connect:Enterprise $$ commands is not enabled. Connect:Enterprise $$ commands, /*SIGNON, and /*BINASC cards embedded in a received batch are treated as data. YES = Scanning for Connect:Enterprise $$ commands is enabled initially, but scanning for a subsequent $$ADD card is not automatic. Each $$ADD card must include the parameter SCAN=YES to continue scanning for $$ commands. Use this value to make FTP command scanning behave like it does in SNA. ALL = Scanning for Connect:Enterprise $$ commands is enabled for the entire batch unless the batch contains a $$ADD card with the parameter SCAN=NO. Use this value to make FTP command scanning behave like it does in BSC.
SECONDARY = 1–16777215	Specifies the secondary allocation amount of the batch to be received. When Connect:Enterprise receives this information, it stores it with the batch information. No other processing is done. There is no default value for this parameter.
TO=NO | YES	TO=YES specifies that the batch being received can only be transmitted once and is not extractable. After transmission, the batch is flagged as non-transmittable and not extractable. If the transmission fails after one or more records have been transmitted, the batch is still marked non-transmittable and not extractable. TO=YES can be thought of as “transmit once, transmit only“. TO=NO specifies that the batch is not marked non-transmittable. The default for this parameter is TO=NO.
TRACKS	Specifies that the primary and secondary allocation amounts of the batch being received are in TRACKS. When Connect:Enterprise receives this information, it will store it with the batch information. No other processing is done. There is no default value for this parameter.

Parameter	Description
TTIME=<to_time>	Specifies the latest creation date and time for a batch to be eligible for processing. Valid formats are: nnn—date calculated as nnn days prior to the current date. hhmm—time expressed in universal (also known as military) time format. For example, 11.15 PM is 22.15. When no date is specified, the date is 1980/001. yyddd—2-digit year and 3-digit Julian day. ccyyddd—2-digit century indicator (19 or 20), 2-digit year, and 3-digit Julian day. nnn:hhmm—number of days prior to current date and time expressed in universal time format. yyddd:hhmm—2-digit year, 3-digit Julian day, and time expressed in universal time format. ccyyddd:hhmm—2-digit century indicator (19 or 20), 2-digit year, 3-digit Julian day, and time expressed in universal time format. There is no default value for this parameter. **Note:** You cannot use two TTIME parameters to set time, for example, TTIME=1100 and TTIME=0134. These two separate parameters taken together do not indicate 11:00 on the 341st day of 2001. To indicate that time, you would use the *yyddd:hhmm* format to specify TTIME=01341:1100.
VBQ#=nn	Specifies the number of the VBQ file on which batches collected in FTP sessions are to be stored. To set this value for multiple collections, the command is: LOCSITE VBQ#=nn KEEP KEEP must be the next parameter after VBQ#=nn to retain the value for more than one batch collection unless it part of an MGET command. If nn is 00, the assignment resets to the Current Collect VBQ. If nn is 01 through 20 that particular VBQ is allocated. **Note:** You cannot display or update this parameters in the ISPF and CICS screens related to the Options Definition File. Only a SITE or LOCSITE command can direct FTP batches to a VBQ other than the Current Collection VBQ.
XMIT=NO | YES	XMIT=NO specifies that the batch being received is only available to the host site and is not available to be transmitted to other sites. XMIT=YES specifies that the batch is available to the host site and is available for transmission to any remote site. The default for this parameter is XMIT=NO.

Examples: “LOCSITE BCHSEP=OPT3 BLOCKS XMIT=NO”

LOCSTAT

Format: LOCSTAT

LOCST

Description: Displays the current settings of Connect:Enterprise -specific parameters set by the LOCSITE command. These parameters affect the GET and MGET commands that add batches, the PUT and MPUT commands that send batches, and the LOCDIR command that lists batches.

The settings displayed in the following messages reflect the parameter values for the remote server definition in use by the FTP Auto Connect session:

```
011-Connect:Enterprise at 10:15:21 on 2003.275 host time.
011-Session started at 10:15:18 on 2003275  ho st time.
011-User: RMTSRV     Current working Mailbox ID: RMTSRV
011-TYPE: A       MODE: S           STRUcture: F
011-Local SITE option values:
011- Allocation type=NONE      BCHSEP=NONE  BLKSIZE=0
011- DIR_FILTER=D                       DIRECTORY=0         DIRFORM=MBOX_ZOS
011- EO=NO   FTIME=1980001:0000   LRECL=0
011- MULTXMIT=NO   ONEBATCH=NO   ORIGIN=             PRIMARY=0
011- RECFM=        REMOTE_FILENAME_LENGTH=LONG     SECONDARY=0
011- TO=NO   TTIME=               XMIT=NO      VBQ#=01   SCAN=NO
011-       606  bytes received for         1 batches during this session
011-         0 Kbytes sent from            0 batches during this session
011-Security Values(SSL):
011   AUTH=SSL  PROT=P  PBSZ=00000
```

In addition to the following LOCSITE parameter setting information, the time the session started is shown (in hh:mm:ss and yyyy.ddd format) along with the number of bytes and batches received and sent during the session. The following table lists the information for each data label in alphabetical order:

Data Label	Description
Allocation type	The type of allocation used – BLOCKS, CYLINDER, TRACKS or NONE.
AUTH	Identifies the security mechanism used (SSL or TLS).
BCHSEP	The batch separation option used – NONE, OPT3, or OPT4 option (see page 243).
BLKSIZE	The file block size.
Current working Mailbox ID	The name of the current working mailbox.
DIR_FILTER	The filter used to exclude batches from the list (see page 244).
DIRECTORY	The file directory size.
DIRFORM	The format of a line written by the LOCDIR command (see page 245).
EO	Indicates if the file can be extracted once (EO) and transmitted (Yes or No).
FTIME	The date in year, month, day format followed by a semicolon and time in hours and minutes representing the from time parameter.
LRECL	The file logical record length.
MODE	The mode – blocked (B), compressed (C), or stream (S).

Data Label	Description
MULTXMIT	Indicates if the batch being received cannot be transmitted multiple times (Yes or No).
ONEBATCH	Indicates if only the first eligible batch can be selected for transfer to the remote FTP client (Yes or No).
ORIGIN	The name of the remote or user ID of the batch job that added the batch.
PBSZ	The size of the protection buffer.
PRIMARY	The file primary allocation amount.
PROT	Indicates the type of data channel protection used by client and server during transfer. (P indicates all data on the channel is encrypted.)
RECFM	The file record format.
REMOTE_ FILENAME_ LENGTH	The length of the Batch ID returned by NLST (short or long).
SCAN	The last known value of the remote (server-side) SCAN setting, which indicates whether the Connect:Enterprise FTP client is scanning RETR received batches for $$ commands and /* cards.
SECONDARY	The file secondary allocation amount.
STRUcture	The file structure – file (F) or record (R).
TO	Indicates if the file can only be transmitted once (TO) and cannot be extracted (Yes or No).
TTIME	The date in year, month, day followed by a semicolon and time in hours and minutes representing the to time (TTIME) parameter.
TYPE	The file type – ASCII (A), EBCDIC (B), or BINARY (I).
USER	The name of the user ID currently logged into the remote FTP Server.
VBQ#	Specifies the VBQ on which batches collected in FTP sessions are to be stored.
XMIT	Indicates if the batch is available to be transmitted to other sites (Yes or No).

Examples: “LOCSTAT”

LS

Format: LS [<foreignfile>]

where <foreignfile> specifies the file names on the remote FTP server to list.

Description: Obtains a list of files on the remote FTP server. The LS command sends the NLST server command. When it is executed, a batch is added to the currently working Mailbox ID with a Batch ID of “NLST + first 19 bytes of <foreignfile>.” This batch contains the list of file names on the remote FTP server. If you omit <foreignfile>, the Batch ID of the added batch is just NLST.

The remote FTP server defines the syntax requirements of the <foreignfile> parameter and the format of the returned list. If the remote FTP server is Connect:Enterprise , use </MID/BID> or <BID> for the <foreignfile> parameter.

By default, the LS command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV ODF parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: “LS /MBOXA/#0000034”—lists batch 34 in the remote server MBOXA mailbox.

“LS /MBOXA”—lists all files in the remote server MBOXA mailbox.

“LS /MBOXA/INVOICE*”—lists all files that begin with INVOICE in the remote server MBOXA mailbox.

‘LS “Order A” “Company A Data” “Company B Data” ’—lists the Order A, Company A Data, and Company B data files in the current working directory on the remote server.

MDELETE

Format: MDELETE <foreignfile> [<foreignfile> … <foreignfile>]

MD <foreignfile> [<foreignfile> … <foreignfile>]

where <foreignfile> specifies the file name on the remote FTP server to delete. You can specify a maximum of 16 file names, separated by blanks. A minimum of one <foreignfile> parameter must be specified.

Description: Deletes one or more files on the remote FTP server.

The MDELETE command has one difference from multiple DELETE commands. If the DELETE command generated by MDELETE fails, no more files are deleted.

When MDELETE or DELETE is issued for a generic filename, the remote server deletes all qualifying files.

A remote Connect:Enterprise server does not physically delete batches but flags all qualifying batches with status flag D, making them unavailable for transmission.

The remote FTP server defines the syntax requirements of the <foreignfile> parameter and the format of the returned list. If the remote FTP server is Connect:Enterprise , use </MID/BID> or <BID> for the <foreignfile> parameter.

The MDELETE command sends a DELETE command to the remote FTP server for each filename; no PORT or NLST commands are sent to the remote server.

Examples: “MDELETE /MBOXA/INVOICE* /MBOXB/RECEIPT*”—deletes all files beginning with INVOICE or RECEIPT from the remote server MBOXA mailbox.

“MDELETE INVOICE* RECEIPT*”—deletes all files beginning with INVOICE or RECEIPT from the current working directory on the remote server.

MGET

Format:

MGET <foreignfile> [<foreignfile> … <foreignfile>]

MG <foreignfile> [<foreignfile> … <foreignfile>]

where <foreignfile> specifies the file name on the remote FTP server to retrieve. You can specify up to a maximum of 16 file names. A minimum of one file name must be specified.

Description:

Collects one or more files from a remote FTP server and adds each file as a batch into the current working mailbox. (You can set the current working mailbox with the LOCCD command. You cannot specify the Mailbox ID in the MGET command.)

The remote FTP server defines the syntax requirements of the <foreignfile> parameter. If <foreignfile> contains a blank character, enclose the entire <foreignfile> in single quotes (’). If <foreignfile> contains an apostrophe, then enclose the entire <foreignfile> in double quotes (“).

The MGET command sends an NLST command to the remote FTP server for each file name in the parameter list. Each entry in the list returned by the remote FTP server NLST command builds RETR commands that are sent to the remote FTP server. The Batch ID of each added batch is created from the last 1–64 characters of the file name and file extension of the foreign file name.

When the MGET command is issued, Connect:Enterprise first issues a type A (ASCII) command. If the user-specified data representation type is not ASCII, Connect:Enterprise remembers the user-specified value and restores it after the MGET completes and before any further processing occurs.

Several LOCSITE parameters affect MGET processing and file characteristics. For more information, see page 238.

By default, the MGET command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

The MGET command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Example 1:

The following MGET command sample retrieves two files named “data.jan” and “data.feb” from the current working directory at the remote FTP server and stores them as two batches in the current working mailbox under the Batch IDs “data.jan” and “data.feb”.

```
“MGET data.jan data.feb”
```

Example 2:

The following MGET command retrieves all files whose name starts with ’partner-Jake-one-for-mailboxA’ from the remote directory path ’/finance/accounting’. Connect:Enterprise FTP stores each file as a batch in the local current working mailbox.

```
“MGET /finance/accounting/partner-Jake-one-for-mailboxA*”
```

The following is a list of the file names returned as a result of the NLST issued by the MGET command:

```
partner-Jake-one-for-mailboxA.011598
partner-Jake-one-for-mailboxA.013198
partner-Jake-one-for-mailboxA.031598
partner-Jake-one-for-mailboxA.033198
```

The MGET command then issues a RETR for each file name returned. The data retrieved from the remote FTP server for each RETR command is added as a batch in the current working mailbox. The Batch ID of each batch is set to the last 1–64 characters of the file name and file extension, as shown in the following:

```
-one-for-mailboxA.011598
-one-for-mailboxA.013198
-one-for-mailboxA.031598
-one-for-mailboxA.033198
```

MKDIR

Format: MKDIR <directory> | < /path/…/directory>

MKD <directory> | < /path/…/directory>

MK <directory> | < /path/…/directory>

where <directory> specifies the name of the file directory to be created at the remote FTP server.

</path/…/directory> specifies the remote server naming convention for the directory to be created.

Description: Creates a new current working directory at the remote FTP server. File transfers are done to and from the current working directory. The remote FTP server defines the syntax requirements of <directory>.

Connect:Enterprise for z/OS Server does not support this command.

Examples: “MKDIR INVOICE”Creates a directory named INVOICE on the remote server.

“MKDIR /VENDOR/Invoice/March”Creates the March subdirectory in the /VENDOR/Invoice directory on the remote server.

MODE

Format: MODE <B | C | S>

MO <B | C | S>

where <B | C | S> specifies the one-character transfer mode:block, compressed, or stream.

Description: Sets the transfer mode for all subsequent data transfers. The data transfer mode stays in effect until the next MODE command is encountered.

In block mode (B), data is transmitted as a series of data blocks, preceded by one or more header bytes. Block mode preserves the logical record boundaries of the data set or file.

In compressed mode (C), data is transmitted as a series of data blocks, preceded by one or more header bytes. Compressed mode preserves the logical record boundaries of the data set or file. In compressed mode, data is transmitted without repetitive characters and blanks. Since additional processing time is required for both the sender and receiver to compress or decompress the data, you should consider the trade-offs before you compress a file.

Note: If MODE is B or C you cannot set STRU=R.

In stream mode (S, the default), data is transmitted as a stream of bytes.

Examples: "MODE s"—sets the block mode as stream.

"MODE" datamode—sets the block mode to whatever the &DATAMODE variable is defined as in the ODF

MPUT

Format: MPUT </MID/BID> | <BID> [</MID/BID> | <BID> … …]

MP </MID/BID> | <BID> [</MID/BID> | <BID> … …]

where </MID/BID> | <BID> specifies the Batch ID of the batches to send to the remote FTP server.

[</MID/BID> | <BID …>] specifies additional batches on the local host to send to the remote FTP server. You can specify up to a maximum of 16 Batch IDs. Separate Batch IDs with a space. At least one </MID/BID> | <BID …> must be specified.

Description: Sends one or more batches from the current working mailbox or from the Mailbox ID specified in </MID/BID> to the remote FTP server. (You can use the LOCCD command to set the current working Mailbox ID.)

The MPUT command searches the current working mailbox or the specified Mailbox ID and sends a STOR or STOU command to the remote FTP server for each matching Batch ID. This is repeated for each </MID/BID> | <BID> parameter in the command. Which command is used (STOR or STOU) depends on the setting of the SUNIQUE command. For more information on the SUNIQUE command, see page 276.

Several LOCSITE parameters affect MGET processing and file characteristics. For more information, see page 238. These parameters can be set using the LOCSITE command and displayed using the LOCSTAT command. For more information on the LOCSTAT command, see page 249.

If SENDSITE is specified in the remote definition, the MPUT command issues a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx command before issuing each STOR or STOU command. The values of LRECL, BLKSIZE and RECFM will be those stored for the batch. If no values are available, the SITE command is not issued.

By default, the MPUT command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

The MPUT command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Batch Selection: If the specified Batch ID is in the format ’#nnnnnnn’, ’nnnnnnn’, ’BID.#nnnnnnn’, or ’#nnnnnnn.dat’ and the batch is in the current working mailbox, the specified batch is eligible for transmission.

If the specified Batch ID is in the format “generic-name”, ’specific’, or ’spec*?[]’, all batches in the current working mailbox that match the Batch ID are eligible for transmission.

If the Batch ID is not specified, all batches in the current working mailbox are eligible to be listed.

Eligible batches are selected depending on the current settings of the FTIME, DIR_FILTER, ORIGIN, and TTIME parameters of the LOCSITE command. An eligible batch is sent if the following conditions are met:

- The batch is contained in an online VBQ.
- The BATCH’S status flags do not match any of the values specified in the DIR_FILTER parameter.
- The batch’s creation date and time is on or after the value specified in the FTIME parameter.
- The batch’s creation date and time is on or before the value specified in the TTIME parameter.
- The batch’s origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.

If the Batch ID is in the format ’#nnnnnnn’, ’nnnnnnn’, ’BID.#nnnnnnn’, or ’#nnnnnnn.dat’, the FTIME, DIR_FILTER, ORIGIN, and TTIME parameters are ignored.

Batch Transmission: If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive intermediate reply. Data is transferred through the data connection.

When you use the STOU command, the remote FTP server assigns the file name.

When you use the STOR command, the MPUT command assigns the file name and places it on the STOR command. The foreign file name format depends on the current setting of the BCHSEP and REMOTE_FILENAME_LENGTH parameters of the LOCSITE command.

If the value of the ONEBATCH parameter is set to YES, the MPUT command selects and transfers only the first eligible batch in each specified path. If the value is set to NO, the MPUT command selects and transfers all available batches in each specified path.

Batch Status Flags:

If BCHSEP=NONE, all eligible batches are concatenated and sent as a single file. As the transmission progresses, each batch is flagged as transmitted immediately after being sent as part of the concatenated file. If a failure occurs during the transmission, only the batches actually sent are flagged as transmitted. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

If BCHSEP=OPT3, all eligible batches are concatenated and sent as a single file. All of the batches are flagged as transmitted only after the entire transmission completes. If a failure occurs during transmission, none of the batches are flagged as transmitted. If the transmission is restarted, all of the batches will be sent.

If BCHSEP=OPT4, each eligible batch is sent as a separate file, and flagged as transmitted immediately after being sent. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

Note: If a batch is flagged as multi-transmittable, it will always be available for subsequent transmission, regardless of the specified BCHSEP value.

Unique File: Names:

The MPUT command syntax provides the capability to specify up to 16 </MID/BID> | <BID> parameters as selection criteria but does not support customer-defined <foreignfile> names. As a result, multiple batches can be selected that have the same Batch ID. When STOU is specified, the FTP remote system generates the unique file name when it saves the file.

When STOR is specified, the system generates unique file names for these batches as follows:

Condition	Description
BCHSEP = OPT4	Each batch is stored as a separate file on the remote FTP server. The system generates unique file names because the Batch ID may be the same for some batches. The file name format used is determined by the REMOTE_FILENAME_LENGTH ODF parameter.
REMOTE_FILENAME_LENGTH = LONG	All file names are generated using the format <BID.#nnnnnnn>, where BID is the Batch ID and nnnnnnn is the 7-digit batch number. Duplicate file names are not generated because the batch number is part of the file name. The file name generated may not be compatible with some file systems.
REMOTE_FILENAME_LENGTH = SHORT	All file names are generated using the format <#nnnnnnn.dat>, where nnnnnnn is the 7-digit batch number. Duplicate file names are not generated because the batch number is part of the file name.

Condition	Description
BCHSEP = NONE /OPT3	Batches are appended together and stored as a single file. A STOR command is issued for each parameter on the MPUT command. The foreign file name generated is the BID of the first batch selected for transmission. Do not use MPUT if this is not acceptable; PUT should be used to allow specific assignment of foreign file names.

Examples: "MPUT BID_A BID_B"Retrieves all batches whose Batch IDs are 'BID_A' and 'BID_B' and sends them to the remote FTP server.

The following paragraphs explain how the BCHSEP options affect this example. Assume that there are five batches that meet the selection criteria. Batch ID BID_A has three batches and Batch ID BID_B has two batches.

- If BCHSEP is not OPT4, two STOR commands with two different file names are created to hold the batches. The first command created is STOR BID_A. Batches one, two, and three are sent to the remote FTP server where they are assigned the file name BID_A. The second command created is STOR BID_B. Batches four and five are sent to the remote FTP server where they are given the file name BID_B.
- If BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=LONG is specified, the batches are sent individually to the remote FTP server using five STOR commands, creating five unique files on the remote FTP server. The commands sent are STOR BID_A.#0000001, STOR BID_A.#00000002, STOR BID_A.#0000003, STOR BID_B.#0000004, and STOR BID_B.#0000005.
- If BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=SHORT is specified, the batches are sent individually to the remote FTP server using five STOR commands, creating five unique files on the remote FTP server. The commands sent are STOR #0000001.dat, STOR #00000002.dat, STOR #0000003.dat, STOR #0000004.dat, and STOR #0000005.dat.

NLST

Format: NLST [<foreignfile>]

NL [<foreignfile>]

where <foreignfile> specifies the file names on the remote FTP server to list.

Description: Obtains a list of file names on the remote FTP server. When NLST is executed, a batch is added to the current working mailbox with a Batch ID of "NLST + first 19 bytes of <foreignfile>." The batch contains a list of file names on the remote FTP server. If you omit <foreignfile>, the Batch ID of the added batch is just NLST.

The remote FTP server defines the syntax requirements of the <foreignfile> parameter and the format of the returned list. If the remote FTP server is Connect:Enterprise , use </MID/BID> or <BID> for the <foreignfile> parameter.

When the NLST command is sent to the FTP server, Connect:Enterprise first issues a TYPE A (ASCII) command. If the user-specified data representation type is not ASCII, Connect:Enterprise remembers the user-specified value and restores it after the DIR completes and before any further processing occurs.

By default, the NLST command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV ODF parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: “NLST /MBOXA/#0000034”—lists batch 34 on the remote server MBOXA mailbox.

“NLST /MBOXA”—lists all files in the remote server MBOXA mailbox.

“NLST /MBOXA/INVOICE*”—lists all files that begin with INVOICE in the remote server MBOXA mailbox.

‘NLST “Order A” “Company A Data” “Company B Data”’—lists the remote server Order A, Company A Data, and Company B Data files.

NOOP

Format: NOOP

Description: Elicits a positive completion reply from the remote FTP server. Any parameters are ignored by the autoclient and sent to the server without an edit. The remote server can accept or reject the command.

Examples: “NOOP”

OPEN

Format: OPEN <ipaddress>[,<port# | 21>]

O <ipaddress>[,<port# | 21>]

where <ipaddress> specifies the internet address of the remote FTP server or firewall server. This can be represented by the standard dotted quad notation, such as 255.255.0.0.

<port# | 21> specifies the host port number to route the Telnet open request. The range is 1-65535.

Description: Presents the TELNET open connection request to the FTP host or a host acting as a firewall for either the FTP client or the FTP host. The Internet address can be presented as a domain name to be resolved into the numeric address used in the final TELNET open.

After logon completes, any subsequent OPEN commands are routed to the server. This enables some proxy servers to function and issue OPEN commands to the FTP server.

Enclose the OPEN command and any case-sensitive values in quotes. For example, ‘OPEN’, ipaddr‘,’portno.

Examples: “OPEN 10.20.129.2,5575”—opens IP address 10.20.129.2 and port 5575.

“OPEN” ipaddr “,” portno—opens the IP address and port number defined in the &IPADDR and &PORTNO variables in the ODF.

"OPEN mvsa, 5575"—opens the domain name mvsa and port 5575.

PASS

Format: PASS <password> [</newpass/newpass>]

PA <password> [</newpass/newpass>]

where <password> specifies the current password for the user to be verified by the FTP server. The <password> cannot contains blanks and must consist of alphabetic characters and numerics. The password can be any length required by the remote FTP server.

</newpass/newpass> specifies a new password, if the FTP server is requested to change the current password. The current password is followed by a slash, the new password, a second slash and a repetition of the new password.

Description: Sends a password to the remote FTP server. The PASS command presents the TELNET password usually required by the FTP server or a firewall.

You can specify a new password with the optional newpass parameter. This command may not be required or supported by all FTP servers and a request to change a password may not be accepted by all servers. The Connect:Enterprise server supports this feature if it is enabled by the host system security operation.

Examples: "PASS mypasswd"—sends the password mypasswd to the remote FTP server.

"PASS" password "/"newpass "/" newpass—sends the values of the &PASSWORD and &NEWPASS variables, defined in the ODF, to the remote FTP server. The &PASSWORD variable defines the current password. The &NEWPASS variable defines a new password. The slashes separating the newpass variables are literals enclosed in quotes.

PUT

Format: PUT </MID/BID> | <BID> [<foreignfile>]

PU </MID/BID> | <BID> [<foreignfile>]

where </MID/BID> | <BID> specifies the specific Batch ID of the batch to send to the remote FTP server.

<foreignfile> specifies the file name on the remote FTP server where the data will be stored.

Description: Transfers batches from the current working mailbox or the Mailbox ID specified in </MID/BID> to the remote FTP server.

The PUT command searches the current working mailbox or the specified Mailbox ID and sends a STOR or STOU command to the remote FTP server for each matching Batch ID. This is repeated for each </MID/BID> | <BID> parameter in the command. Which command is used (STOR or STOU) depends on the setting of the SUNIQUE command. For more information on the SUNIQUE command, see page 276.

Several LOCSITE parameters affect PUT processing and file characteristics. For more information, see page 238. These parameters can be set using the LOCSITE command and displayed using the LOCSTAT command. For more information on the LOCSTAT command, see page 249.

If SENDSITE is specified in the remote definition, the PUT command issues a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx command before issuing each STOR/STOU command. The values of LRECL, BLKSIZE and RECFM are stored for the batch. If no values are available, the SITE command is not issued.

By default, the PUT command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

The PUT command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Batch Selection: The Batch ID specified in </MID/BID> | <BID> may be specific or generic. All batches in the current working mailbox that match the Batch ID are eligible to be transmitted.

Eligible batches are selected depending on the current settings of the FTIME, ORIGIN, and TTIME parameters of the SITE command and the current data type. An eligible batch is selected for transmission if the following conditions are met:

- The batch is contained in an online VBQ.
- The batch’s creation date and time is on or after the value specified in the FTIME parameter.
- The batch’s creation date and time is on or before the value specified in the TTIME parameter.
- The batch’s origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.

Batch Transmission: If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive intermediate reply. Data is transferred through the data connection.

When the STOU command is used, the remote FTP server assigns the file name and the <foreignfile> parameter is ignored.

When the STOR command is used, the file name is assigned by the <foreignfile> parameter. If the <foreignfile> parameter is not specified, the file name is assigned by the PUT command. The format of the assigned foreign file name depends on the current setting of the BCHSEP and REMOTE_FILENAME_LENGTH parameters of the LOCSITE command.

If you specify ONEBATCH=YES, the PUT command only sends the first batch selected.

Batch Status Flags: If BCHSEP=NONE, all eligible batches are concatenated and sent as a single file. As the transmission progresses, each batch is flagged as transmitted immediately after being sent as part of the concatenated file. If a failure occurs during the transmission, only the batches actually sent are flagged as transmitted. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

If BCHSEP=OPT3, all eligible batches are concatenated and sent as a single file. All of the batches are flagged as transmitted only after the entire transmission completes. If a failure occurs during transmission, none of the batches are flagged as transmitted. If the transmission is restarted, all of the batches will be sent.

If BCHSEP=OPT4, each eligible batch is sent as a separate file, and flagged as transmitted immediately after being sent. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

Note: If a batch is flagged as multi-transmittable, it will always be available for subsequent transmission, regardless of the specified BCHSEP value.

Unique File Names: One or more batches may be selected for transmission. When STOU is used, the unique file name is determined by the remote FTP server. When STOR is used, the method used for creating a unique file name varies depending on the following BCHSEP options:

Condition	Description
BCHSEP = OPT3/NONE	The <foreignfile> parameter will be used as the file name on the remote system. If <foreignfile> is not specified, the Batch ID is used as the file name. If the Batch ID is a wildcard (*) or only a Mailbox ID is specified, then the first Batch ID from the first transmitted batch is used as the file name on the remote system.
BCHSEP = OPT4	Each batch is stored with a unique name. The generated file name on the remote system depends on the setting of the ODF *REMOTES REMOTE_FILENAME_LENGTH parameter.
REMOTE_FILENAME_LENGTH = LONG	The file name on the remote system will be <foreignfile>.#nnnnnnn where nnnnnnn is the batch number of the transmitted Connect:Enterprise . If <foreignfile> is not specified, the file name is BID.#nnnnnnn. If the Batch ID is a wildcard (*) or only a Mailbox ID is specified, then the file name is the first Batch ID from the first transmitted batch followed by .#nnnnnnnn.
REMOTE_FILENAME_LENGTH = SHORT	The file name on the remote system will be #nnnnnnn.dat, where nnnnnnn is the batch number of the transmitted Connect:Enterprise batch. Duplicate file names are not generated because the batch number is part of the file name. Note that the Batch ID is not included in the name. Although this may make the file name less descriptive, it guarantees a unique file name that should be compatible with all file systems.

Example 1: This example assumes that BCHSEP=NONE or OPT3 and SUNIQUE specifies STOR. The following PUT command retrieves the batch named “batch file” from the current

working mailbox on the local host and sends it to the remote FTP server where it is stored as “from.mailboxA”. All batches with the Batch ID “batch file” are concatenated and sent to the remote FTP server as a single set of data.

```
“PUT ’batch file’ from.mailboxA”
```

Example 2: This example assumes that BCHSEP=NONE or OPT3 and SUNIQUE specifies STOR. The following two PUT commands function the same. They both retrieve all of the batches whose Batch ID starts with the characters BID_, and send them to the remote FTP server to be stored as file name “partnerA.dat”.

```
‘PUT “BID_” partnerA.dat’
```

or

```
“PUT ’BID_*’ partnerA.dat”
```

Example 3: This example assumes that BCHSEP=OPT4, REMOTE_FILENAME_LENGTH=LONG, and SUNIQUE specifies STOR.

The following PUT command retrieves all batches named “batch file” from the current working mailbox on the local host and sends them to the remote FTP server where they are stored as separate files.

```
“PUT ’batch file’ from.mailboxA”
```

Assuming that the batches requested from Batch ID “batch file” are batch numbers 26, 27, and 28, the files are stored with the following filenames: “from.mailboxA.#0000026,” “from.mailboxA.#0000027,” and “from.mailboxA.#0000028.”

Example 4: This example assumes that BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=SHORT, and SUNIQUE specifies STOR.

The following PUT command retrieves all batches named “batch file” from the current working mailbox on the local host and sends them to the remote FTP server where they are stored as separate files.

```
“PUT ’batch file’ from.mailboxA”
```

If the batches that can be requested from Batch ID “batch file” are batch numbers 2, 23, and 26, the files are stored with the following filenames: “#000002.dat,” “#0000023.dat,” and “#0000026.dat.” The “from.mailboxA” foreign file name is ignored.

Example 5: This example assumes that BCHSEP=OPT4, REMOTE_FILENAME_LENGTH=SHORT and SUNIQUE specifies STOU.

The following PUT command retrieves all batches named “batch file” from the current working mailbox on the local host and sends them to the remote FTP server, where they are stored as three separate files with names selected by the FTP remote server.

```
“PUT ’batch file’ from.mailboxA”
```

The <foreignfile> name “from.mailboxA” is ignored.

PWD

Format: PWD

Description: Displays the remote FTP server’s current working directory.

Examples: “PWD”

QUIT

Format: QUIT

QUI

Description: Immediately terminates the server connection. The script continues processing but the connection to the server is terminated.

Examples: “QUIT”

QUOTE

Format: QUOTE <string>

QUO <string>

where <string> specifies the information to be sent to the remote FTP server.

Description: Sends a user specified string, as is, to the server for execution.

Examples: “QUOTE testing123”Sends the string ‘testing123’ to the server.

RETR

Format: RETR <foreignfile> </MID/BID> | <BID>

RET <foreignfile> </MID/BID> | <BID>

where <foreignfile> specifies the name of the file to be retrieved from the remote FTP server.

</MID/BID> specifies the non-current working Mailbox ID and the Batch ID of the batch to be added.

<BID> specifies the Batch ID of the batch to add to the current working mailbox.

Description: Collects a file from a remote FTP server and adds it to the current working mailbox (or non-current working mailbox if </MID/BID> used).

The <foreignfile> parameter identifies the file at the remote FTP server. The remote FTP server defines the syntax requirements of the <foreignfile> parameter. If the remote FTP server is Connect:Enterprise :

- Specify </MID/BID> to collect a file from the non-current working mailbox
- Specify <BID> when collecting from the current working mailbox

Any data specified after the second parameter is ignored.

The Mailbox ID specified in </MID/BID> overrides the setting of the current working Mailbox ID for this command, but does not change the current working Mailbox ID value.

Several LOCSITE parameters affect RETR processing and file characteristics. For more information, see page 238.

By default, the RETR command sends the FTP server a PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

Examples: "RETR /XYZ/BillData /MBOXB/PhoneBill"—retrieves Batch ID BillData from mailbox XYZ from the remote Connect:Enterprise FTP server and adds it to mailbox MBOXB with Batch ID PhoneBill on the local Connect:Enterprise client.

"RETR InvoiceData /XYZ/XYZInvoice"—retrieves the InvoiceData file from the remote FTP server and adds it to mailbox XYZ with a Batch ID of XYZInvoice.

"RETR InvoiceData XYZInvoice"—retrieves the InvoiceData file from the remote FTP server and adds it to the current working mailbox with a Batch ID of XYZInvoice.

"RETR InvoiceData /XYZ/'March Invoice'"—retrieves the InvoiceData file from the remote FTP server and adds it to the mailbox XYZ with a Batch ID of 'March Invoice'.

RMDIR

Format: RMDIR <directory>

RMD <directory>

RM <directory>

where <directory> specifies the name of the directory to be removed from the remote FTP server.

Description: Removes a directory on the remote FTP server. Connect:Enterprise for z/OS Server does not support this command.

Examples: "RMDIR INVOICE"—removes the directory INVOICE from the remote FTP server.

"rmd MARCH"—removes the directory MARCH from the remote FTP server.

SCGET

Format: SCGET [</MID/BID> | <BID>]

where </MID/BID> | <BID> specifies the Batch ID of the batches to retrieve.

Description: Retrieves batches from a remote Connect:Enterprise and adds them to the local Connect:Enterprise under the same Mailbox ID of the retrieved batch.

The SCGET command differs from the GET and MGET commands in that SCGET copies the Batch IDs from the remote Connect:Enterprise files to the new files on the local Connect:Enterprise . The new files on the local Connect:Enterprise are assigned unique batch numbers.

When the NLST command is send to the FTP server, Connect:Enterprise first issues a TYPE I (Binary) command. If the user-specified data representation type is not ASCII, Connect:Enterprise remembers the user-specified value and restores it after the DIR completes and before any further processing occurs.

By default, the SCGET command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV ODF parameter is specified for the remote, the FTP server PASV command is sent instead. This provides the ability to send and receive data through a firewall.

SCGET requires that the LOCSITE parameter IDENT be set to yes (the default).

When using SCGET, the IDENT parameter in the *REMOTES section of the ODF located on the server end of the connection must be set to YES.

The SGET command enforces BCHSEP=04, TYPE=I, REMOTE_FILE_NAME=SHORT, MODE=B, and STRU=F.

If any transfer fails, the SCGET command ends and all remaining transfers are not attempted.

Example 1: The following commands retrieve all batches with Batch ID BID_A from the remote Connect:Enterprise mailbox BIDBOX and add them under the Mailbox ID BIDBOX. Assume three batches in this example.

```
"CD BIDBOX"
"SCGET BID_A"
```

The result is three batches added to the mailbox with a Batch ID of BID_A and with batch numbers unique to the local Connect:Enterprise .

Example 2: The following command retrieves all batches starting with Batch ID BID_ from the remote Connect:Enterprise current working mailbox BIDBOX and adds them to the local Connect:Enterprise under the same Mailbox ID as the remote mailbox. Assume there are three batches in BID_A, two in BID_B, and one in BID_C.

```
"SCGET BID_*"
```

The result is six batches added to local Connect:Enterprise under Mailbox ID BIDBOX–three with a Batch ID of BID_A, two with a Batch ID of BID_B, and one with a Batch ID of BID_C. Each batch has a batch number unique to the local Connect:Enterprise .

SCPUT

Format: SCPUT </MID/BID> | <BID>

where </MID/BID> | <BID> specifies the Batch ID of the batches to send.

Description: When using SCPUT, the IDENT parameter in the *REMOTES section of the ODF located on the server end of the connection must be set to YES.

Transfers batches from a local Connect:Enterprise current working mailbox or a specified Mailbox ID to a remote Connect:Enterprise current working mailbox. A transferred batch uses the same Mailbox ID and Batch ID on both systems.

The SCPUT command differs from the MPUT and PUT commands in that SCPUT assigns the Mailbox ID and Batch ID used on the local Connect:Enterprise to the batches added to the remote Connect:Enterprise . Batches added to the remote Connect:Enterprise are assigned unique batch numbers.

The SCPUT command searches the local Connect:Enterprise current working mailbox or the specified Mailbox ID and sends a TYPE, STRU, MODE, SITE, and STOR command to the remote FTP server for each matching Batch ID.

The EO, MULTXMIT, TO, and XMIT parameters of the remote FTP server SITE command do not affect the batches added to the remote Connect:Enterprise using SCPUT.

The SCPUT command issues a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx STATFLGS=‘x’ command before issuing each STOR command. The values of LRECL, BLKSIZE, and RECFM are those stored for the batch. If no values are available, the SITE command uses only the STATFLGS parameter.

By default, the SCPUT command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote site, the FTP server PASV command is sent instead. This parameter provides the ability to send and receive data through a firewall.

Command SCPUT requires that LOCSITE parameter IDENT be set to YES. YES is the default for IDENT.

BCHSEP uses OPT4 with the SCPUT command. The <BID> parameter of the STOR command is set to the Batch ID of the batch being transferred. The REMOTE_FILENAME_LENGTH parameter does not modify the file name format for SCPUT.

The SCPUT command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Batch Selection: If the specified Batch ID is in the format '#nnnnnnn', 'nnnnnnn', 'BID.#nnnnnnn', or '#nnnnnnn.dat' and the batch is in the current working mailbox, the specified batch is eligible for transmission.

If the specified Batch ID is in the format "generic-name", 'specific', or 'spec*?[]', all batches in the local Connect:Enterprise current working mailbox that match the Batch ID are eligible for transmission.

Eligible batches are selected depending on the current settings of the FTIME, DIR_FILTER, ORIGIN, and TTIME parameters of the LOCSITE command and the current data type. The batch is contained in an online VBQ.

- The batch status flags match the current data type.
- The batch's creation date and time is on or after the value specified in the FTIME parameter.
- The batch's creation date and time is on or before the value specified in the TTIME parameter.
- The batch's origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.
- The batch's status flags match the value of the DIR_FILTER parameter, or the DIR_FILTER parameter is blank.

If the Batch ID is in the format '#nnnnnnn', 'nnnnnnn', 'BID.#nnnnnnn', or '#nnnnnnn.dat', the FTIME, DIR_FILTER, ORIGIN, and TTIME parameters are ignored.

The FTIME, ORIGIN, DIR_FILTER, and TTIME parameters of the LOCSITE command and the TYPE command affect SCPUT processing. The FTIME, ORIGIN, DIR_FILTER, and TTIME parameters can be set using the LOCSITE command and displayed using the LOCSTAT command. For more information on the LOCSITE command, see page 238. For more information on the LOCSTAT command, see page 249.

Batch Transmission: If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive intermediate reply. Data is transferred through the data connection.

Batch Status Flags: Each batch is flagged as transmitted when it is sent successfully.

Example 1: The following command sends batches in Batch ID BID_A from a local Connect:Enterprise current working mailbox to a remote Connect:Enterprise :

```
"SCPUT BID_A"
```

Assuming three batches, the batches are sent using the following three STOR commands:

```
"STOR BID_A"
"STOR BID_A"
"STOR BID_A"
```

Example 2: The following command sends batches in any Batch ID beginning with BID_ from a local Connect:Enterprise mailbox to a remote Connect:Enterprise :

```
"SCPUT BID_*"
```

Assuming that there are three batches in BID_A, two in BID_B, and one in BID_C, the batches are sent using the following six FTP server STOR commands:

```
"STOR BID_A"
"STOR BID_A"
"STOR BID_A"
"STOR BID_B"
"STOR BID_B"
"STOR BID_C"
```

SITE

Format: SITE [<string>]

SI [<string>]

where <string> specifies the information to send to the remote FTP server.

Description: Sends site-specific information to the remote FTP server. The value in the <string> parameter depends on the SITE commands accepted by the remote FTP server. If the remote FTP server is a Connect:Enterprise system, see the description of the SITE command in the *Connect:Enterprise for z/OS Remote User's Guide* for supported SITE parameters.

Examples: "SITE BCHSEP=OPT3"—sends the SITE BCHSEP parameter to a remote Connect:Enterprise FTP server.

STATUS

Format: STATUS

STAT

STA

Description: Retrieves site-specific status and setting information from the remote FTP server.

The type of information returned depends on the remote server. If the remote FTP server is a Connect:Enterprise system, the status and setting information is displayed in the following messages to the remote FTP client:

```
211-Connect:Enterprise at 10:09:49 on 2003.275 host time.
211-Session started at 10:09:33 on 2003/275 host time.
211-User: FTPRMT     Current working Mailbox ID: FTPRMT
211-TYPE: A       MODE: S           STRUcture: F
211-Local SITE option values:
211- Allocation type=NONE      BCHSEP=NONE  BLKSIZE=0
211- DIR_FILTER=18Fe                    DIRECTORY=0         DIRFORM=MBOX_ZOS

211- EO=NO   FTIME=1980001:0000   LRECL=0      LS_FILTER=BDI!RST

211- MULTXMIT=NO   ONEBATCH=NO   ORIGIN=             PRIMARY=0
211- RECFM=        REMOTE_FILENAME_LENGTH=LONG     SECONDARY=0
211- TO=NO   TTIME=               XMIT=NO       VBQ#=01  SCAN=NO
211-         0 Kbytes received for         0 batches during this session
211          0 Kbytes sent from            0 batches during this session
```

Examples: “STATUS”

“STAT”

STOR

Format: STOR </MID/BID> | <BID> [<foreignfile>]

STO </MID/BID> | <BID> [<foreignfile>]

where </MID/BID> | <BID> specifies the Batch ID of the batch to be sent to the remote FTP server.

<foreignfile> specifies the file name on the remote FTP server where the data will be stored.

Description: Transfers batches from the current working mailbox or the specified Mailbox ID to the remote FTP server. (You can set the current working mailbox with the LOCCD command.)

The STOR command searches the current working mailbox or the specified Mailbox ID for all batches that match the specified Batch ID. A STOR command is sent to the remote FTP server for each matching Batch ID.

The difference between the STOR and STOU commands is that you can specify a <foreignfile> with STOR, but not with STOU.

Several LOCSITE parameters affect STOR processing and file characteristics. For more information, see page 238. These parameters can be set using the LOCSITE command and displayed using the LOCSTAT command. For more information on the LOCSTAT command, see page 249.

If SENDSITE is specified in the remote definition, the STOR command issues a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx command before issuing the STOR command. The values of LRECL, BLKSIZE and RECFM will be those stored for the batch. If no values are available, the SITE command is not issued.

By default, the STOR command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server

PASV command is sent instead. This provides the ability to send and receive data through a firewall.

The STOR command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Batch Selection:

The Batch ID specified in </MID/BID> | <BID> may be specific or generic. All batches in the current working mailbox that match the Batch ID are eligible to be transmitted.

Eligible batches are selected depending on the current settings of the FTIME, ORIGIN, and TTIME parameters of the SITE command and the current data type. An eligible batch is selected for transmission if the following conditions are met:

- The batch is contained in an online VBQ.
- The batch's creation date and time is on or after the value specified in the FTIME parameter.
- The batch's creation date and time is on or before the value specified in the TTIME parameter.
- The batch's origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.

Batch Transmission:

If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive intermediate reply. Data is transferred through the data connection.

If ONEBATCH=YES is specified, the STOR command only sends the first batch selected.

Batch Status Flags:

If BCHSEP=NONE, all eligible batches are concatenated and sent as a single file. As the transmission progresses, each batch is flagged as transmitted immediately after being sent as part of the concatenated file. If a failure occurs during the transmission, only the batches actually sent are flagged as transmitted. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

If BCHSEP=OPT3, all eligible batches are concatenated and sent as a single file. All of the batches are flagged as transmitted only after the entire transmission completes. If a failure occurs during transmission, none of the batches are flagged as transmitted. If the transmission is restarted, all of the batches will be sent.

If BCHSEP=OPT4, each eligible batch is sent as a separate file, and flagged as transmitted immediately after being sent. If the transmission is restarted, only the remaining eligible batches not already marked as transmitted will be sent.

Note: If a batch is flagged as multi-transmittable, it will always be available for subsequent transmission, regardless of the specified BCHSEP value.

Unique File Names: The method used for creating a unique file name varies depending on the BCHSEP option as follows:

Condition	Description
BCHSEP= OPT3/NONE	The file name on the remote system will be <foreignfile>. If <foreignfile> is not specified, the Batch ID will be used as the file name. If the Batch ID is a wildcard (*) or only the Mailbox ID is specified, then the first Batch ID from the first transmitted batch is used as the file name on the remote system.
BCHSEP=OPT4	Each batch is stored with a unique name. The generated file name on the remote system depends on the setting of the ODF *REMOTES REMOTE_FILENAME_LENGTH parameter.
REMOTE_FILENAME_LENGTH= LONG	The file name on the remote system will be <foreignfile>.#nnnnnnn where nnnnnnn is the batch number of the transmitted Connect:Enterprise batch. If <foreignfile> is not specified, the file name is BID.#nnnnnnn. If the Batch ID is a wildcard (*) or only the Mailbox ID is specified, then the file name is the first Batch ID of the first transmitted batch followed by .#nnnnnnnn.
REMOTE_FILENAME_LENGTH= SHORT	The file name on the remote system will be #nnnnnnn.dat where nnnnnnn is the batch number of the transmitted Connect:Enterprise batch.

Example 1: This example assumes that BCHSEP=NONE or OPT3. The following STOR command retrieves the batch named “batch file” from the current working mailbox on the local host and sends it to the remote FTP server where it is stored as ”from.mailboxA”. All batches with the Batch ID “batch file” are concatenated and sent to the remote FTP server as a single set of data.

```
“STOR ’batch file’ from.mailboxA”
```

Example 2: This example assumes that BCHSEP=NONE or OPT3. The following STOR command retrieves the batch named ’batch file’ from the current working mailbox on the local host and sends it to the remote FTP server where it is stored as “batch file” because <foreignfile> is not specified on the STOR command. All batches with the Batch ID “batch file” are concatenated and sent to the remote FTP server as a single set of data.

```
“STOR ’batch file’ ”
```

Example 3: This example assumes that three batches exist with a Batch ID of “batch file” and batch numbers 23, 26, and 29 in the current working mailbox. It also assumes that BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=LONG are specified.

The following STOR command retrieves all batches named ’batch file’ from the current working mailbox on the local host and sends them to the remote FTP server where they are stored as three separate files: “from.mailboxA.#0000023,” “from.mailboxA.#0000026,” and “from.mailboxA.#0000029.”

```
“STOR ’batch file’ from.mailboxA”
```

Example 4: This example assumes that three batches exist with a Batch ID of “batch file” and batch numbers 23, 26, and 29 in the current working mailbox. It also assumes that BCHSEP=OPT4 and REMOTE_FILENAME_LENGTH=SHORT are specified.

The following STOR command retrieves all batches named “batch file” from the current working mailbox on the local host and sends them to the remote FTP server where they are stored as three separate files: “#0000023.dat,” “#0000026.dat,” and “#000002.dat.” The <foreignfile> parameter “from.mailboxA” is ignored.

```
“STOR ’batch file’ from.mailboxA”
```

STOU

Format: STOU </MID/BID> | <BID>

where </MID/BID> | <BID> specifies a specific Batch ID to be sent to the remote FTP server.

Description: Transfers batches from the current working mailbox or the specified Mailbox ID to the remote FTP server. (You can set the current working mailbox with the LOCCD command.)

The STOU command searches the current working mailbox or the specified Mailbox ID for all batches that match the specified Batch ID. A STOU command is sent to the remote FTP server for each matching Batch ID.

The difference between the STOR and STOU commands is that you can specify a <foreignfile> with STOR, but not with STOU.

Several LOCSITE parameters affect STOU processing and file characteristics. For more information, see page 238. These parameters can be set using the LOCSITE command and displayed using the LOCSTAT command. For more information on the LOCSTAT command, see page 249.

If SENDSITE is specified in the remote definition, the STOU command first issues a SITE LRECL=nnnnn BLKSIZE=nnnnn RECFM=xx command. The values of LRECL, BLKSIZE and RECFM will be those stored for the batch. If no values are available, the SITE command is not issued.

By default, the STOU command sends the FTP server PORT command to establish a data connection. If the ODF SENDPASV parameter is specified for the remote, the FTP server

PASV command is sent instead. This provides the ability to send and receive data through a firewall.

The STOU command terminates if a permanent negative completion reply code or an unexpected reply code is encountered.

Batch Selection: The Batch ID specified in </MID/BID> | <BID> must be specific, without wildcards. All batches in the current working mailbox that match the Batch ID are eligible to be transmitted.

Eligible batches are selected depending on the current settings of the FTIME, ORIGIN, and TTIME parameters of the SITE command and the current data type. An eligible batch is selected for transmission if the following conditions are met:

- The batch is contained in an online VBQ.
- The batch's creation date and time is on or after the value specified in the FTIME parameter.
- The batch's creation date and time is on or before the value specified in the TTIME parameter.
- The batch's origination matches the value of the ORIGIN parameter or the ORIGIN parameter is blank.

Batch Transmission: If no batches are selected, the command returns a permanent negative completion reply.

If batches are selected, the command returns a positive intermediate reply. Data is transferred through the data connection.

When you use the STOU command, the remote FTP server assigns the file name and any <foreignfile> parameter specified is ignored.

If ONEBATCH=YES is specified, the STOU command only sends the first batch selected.

Batch Status Flags: If BCHSEP=NONE or BCHSEP=OPT4, each batch is flagged as transmitted when it is sent successfully. If a failure occurs during transmission, BCHSEP=NONE or BCHSEP=OPT4 prevent batches previously sent from being retransmitted. If the transmission is restarted to the same remote client file, the data from the previous batches is replaced with the data from the unsent batches, and the data from the previously sent batches is lost. If the transmission is restarted to a different remote client file, data sent in previous batches is not lost.

If BCHSEP=OPT3, all batches are flagged as transmitted when all batches have been sent successfully. If a failure occurs during transmission, BCHSEP=OPT3 allows all batches to be transmitted again. If the transmission is restarted to the same remote client file, the data from the previous batches is retransmitted and no data is lost. If the transmission is restarted to a different remote client file, duplicate records are sent to the remote site.

Example 1: This example assumes that BCHSEP=NONE or OPT3. The following STOU command retrieves the batch named ’batch file’ from the current working mailbox on the local host and sends it to the remote FTP server where it is stored with a file name determined by the remote FTP server. All batches with the Batch ID “batch file” are concatenated and sent to the remote FTP server as a single set of data.

```
“STOU ’batch file’
```

Example 2: This example assumes that BCHSEP=NONE or OPT3. The following two STOU commands are functionally the same. They both retrieve all batches whose Batch ID starts with the characters “BID_”, and send them to the remote FTP server to be stored with file names determined by the remote FTP server.

```
‘STOU “BID_” ’
```

```
“STOU ’BID_*’ ”
```

Example 3: This example assumes three batches exist with a Batch ID of “batch file” and batch numbers 23, 26, and 29 in the current working mailbox. It also assumes BCHSEP=OPT4 is specified.

The following STOU command retrieves the batches named “batch file” from the current working mailbox on the local host and sends them to the remote FTP server where they are stored as three separate files with names selected by the remote FTP server.

```
“STOU ’batch file’ ”
```

STRUCT

Format: STRU <F | R>

STR <F | R>

where <F | R> specifies one of the following file structure types:

- F–file (default)
- R–record

Description: Sets the transfer data structure for all subsequent data transfers. The data structure stays in effect until the next STRU command is encountered. The following table explains the effect on data transfers:

Structure Value	Effect on Data Transfer
F (file)	Batches received while structure file is in effect are added to the current working mailbox as contiguous data. No attempt is made to convert the structure file into structured records. Batches sent while structure file is in effect are sent as contiguous data.
R (record)	Batches received while structure record is in effect are added to the current working mailbox as a single batch with multiple records. Batches sent while structure record is in effect are sent in multiple records. You cannot use STRU=R if MODE B or C has been set.

Examples: “STR F”—sets structure as file.

“STRU“ datastru—sets structure to the value defined in the ODF &DATASTRU variable

SUNIQUE

Format: SUNIQUE [NO | YES]

SU [NO | YES]

Description: Sets the command that MPUT and PUT send to the remote FTP server to either STOR or STOU.

Specifying NO sets the command sent to STOR.

Specifying YES sets the command sent to STOU.

Examples: “SUNIQUE no”—sets the command sent by MPUT and PUT to the remote FTP server to STOR.

“SUNIQUE”—sets the command sent by MPUT and PUT to the remote FTP server to STOU (Yes being the default).

SYSTEM

Format: SYSTEM

SYST

SY

Description: Sends the SYST command remote FTP server to displays the name and level of the remote FTP server operating system.

Examples: “SYSTEM”

TYPE

Format: TYPE <A | E | I>

TY <A | E | I>

where < A | E | I > specifies the following data representation types:

- A–ASCII (the default)
- E–EBCDIC
- I–Image (Binary)

Description: Specifies the data representation type.

The ASCII data type contains characters that can be displayed and the <CRLF> characters (hexadecimal ‘0D0A’) are used to delimit lines within the data. The EBCDIC data type contains characters that can be displayed and the <NL> character (hexadecimal ‘15’) is used to delimit lines within the data. The binary data type contains a contiguous stream of bits with no line delimiters.

For Connect:Enterprise , data types ’A’, ’A<SP>N’, ’A<SP>T’, and ’A<SP>C’ are identical and are treated as ASCII non-print. Data types ’E’, ’E<SP>N’, ’E<SP>T’, and ’E<SP>C’ are also identical, and are treated as EBCDIC non-print.

Examples: “TYPE a”—sets the data representation type as ASCII.

“TYPE” datatype—sets the data representation type to the value of the ODF &DATATYPE variable.

USER

Format: USER <username>

U <username>

where <username> specifies the user name to be associated with a connection to the FTP server.

Description: Sends an identifying name to the remote FTP server. The USER command presents the TELNET identification required by the FTP server for access to its file system or to allow a firewall to pass the connection to the next server. The <username> parameter normally contains no blanks or special characters and consists of alphabetic characters and numerics.

Examples: “USER pete1”—sets the user name to pete1.

“USER” userid—sets the user name to the value defined in the ODF &USERID variable.

USERLOG

Format: USERLOG [<“ULFC=” nnn “ “ | “,”>] [<string>]

UL [<“ULFC=” nnn “ “ | “,”>] [<string>]

where <“ULFC=” nnn “ “ | “,”> specifies a User Log Fail Code

- ULFC stands for User Log Fail Code
- nnn is any number in the range 240–255

- <string> specifies the text, which can contain up to 480 characters, to be added to the log record

Description: Requests that the FTP client write a log record that includes the specified user log failure code and string of descriptive text. The failure code along with any user-defined text in the string is listed in Auto Connect summary and detail reports and on screens in the ISPF or CICS user interface. The failure code descriptions that are printed in the offline log reports are contained in the STUTAAMT. For more information on how to define user log failure codes in the STUTAAMT table, see the chapter on offline utilities in *Connect:Enterprise for z/OS User's Guide.*

Examples: "USERLOG PASS command failed for USER="userid—writes the string "PASS command failed for USER=userid" where userid is defined in the ODF &USERID variable.

!TIMER

Format: !TIMER <OFF | ON>

where <OFF | ON> specifies to turn on or off the loop/hang timer.

Description: The loop/hang timer is automatically maintained by each client thread to provide recovery from a loop condition which permanently ties up the thread. The timer is turned on when a LOGON_SCRIPT or AC_SCRIPT program begins and turned off when it ends. It is also turned off when an RDXFTPAC host command begins and turned on again when it ends. The RDXFTPAC host commands set their own timers to detect server problems, so they can take recovery action based on the specific circumstances.

The !TIMER OFF command is provided so a program can turn off the timer to prevent it from stopping the script during a long-running, non-RDXFTPAC process. Once off, the timer remains off until either a !TIMER ON command or an RDXFTPAC host command is issued.

If the loop/hang timer is OFF and a program or a subroutine loops or hangs, the only way to end the program is to force the thread.

When the loop/hang timer is ON, the ODF parameter SCRIPT_INTERVAL_TIME sets the maximum number of wall clock seconds that a program can run without issuing an RDXFTPAC host command. After each RDXFTPAC host command is issued, the program gets another full SCRIPT_INTERVAL_TIME interval to perform non-RDXFTPAC host commands. If the interval is exceeded, the timer exit calls IRXIC to raise a REXX HALT condition. In a program that is not trapping the HALT condition, the program immediately ends.

If the program traps the HALT condition, a second timer is set for the same amount of time as the first timer. If the second timer is exceeded, another HALT condition is raised. If this is also trapped, a third timer is set which, if exceeded, attempts to stop the program once more without abending. If the program is stopped by any of these attempts, it is reported in the Connect:Enterprise job log under message CMB2129I CC=0933 and in the VLF log record with fail code 163. If the program still has not stopped, then a fourth timer is set which, if exceeded, stops the program with a User Abend 1057. The Abend is reported in the VLF log record with fail code 157.

Examples: “!TIMER OFF”—turns off the timer while sleeping.

rc = syscalls(‘ON’)—enables the UNIX sleep command.

address syscall “SLEEP 60”—goes to sleep for 60 seconds.

“!TIMER ON”—turns the timer back on.

!WTO

Format: !WTO [<string>]

where <string> specifies the text to be written to the console and JES2 job log.

Description: Requests that the FTP client issue a WTO containing the specified string. The WTO is issued with Route Code 11, Descriptor code 7.

Examples: “!WTO PASS command failed for USER=”userid—issues the WTO string “PASS command failed for USER=userid” where userid is defined in the ODF &USERID variable

Chapter 10

Implementing the Connect:Enterprise for z/OS Security Interface

The Connect:Enterprise security interface enables you to secure the Connect:Enterprise system and the VSAM files, control access to remote communications requests, control access and operations performed on batch files in the repository, and implement some or all of the security functions in phases or at the same time.

Note: Connect:Enterprise for z/OS supports both the TLS (Transport Layer Security) and SSL (Secure Sockets Layer) protocols. Throughout this chapter, the phrase SSL is used to describe both the SSL and TLS protocols.

Security Functions and Requirements for Full Implementation

Securing access to the Connect:Enterprise system and the batches has several components. The following table describes the Connect:Enterprise security functions and tasks. All of the Connect:Enterprise security functions are optional.

Security Function	Task
Define authority under which Connect:Enterprise runs	Define the Connect:Enterprise user/user ID. See *Defining a User/User ID for Connect:Enterprise* on page 283.
Secure the VSAM files from unauthorized access by users outside Connect:Enterprise	◆ Define the user ID for the VSAM file server. ◆ Define data set security rules for the VSAM batch files. See *Defining the User ID for the VSAM File Server Task* on page 284 and *Defining Data Set Security Rules for VSAM Files* on page 286.

Security Function	Task
Control remote access to communications sessions with Connect:Enterprise using the logon security interface	Within your security software, define the user ID (remote site name) and password for each remote site that accesses Connect:Enterprise. See *Controlling Remote Access with the Logon Security Interface* on page 287.
Control the functions invoked by remote, APPC, and offline utility users to process specific batches using the optional batch/function security interface	Create rules based on type of access to verify that the remote user, APPC user, or STOUTL job has UPDATE authority for the generated data set name used for the Connect:Enterprise system and the batch or function being processed. See *Controlling User Processing of Batches with the Batch/Function Security Interface* on page 289.
Implement and activate the security interface	◆ Create a hierarchy of global, system, and individual user security rules. See *Creating the Security Interface Control Hierarchy* on page 292. For prerequisites for using the complete interface, see ◆ Specify the level of security checking in the ODF. See Chapter 3, *Configuring *OPTIONS Record for System Resources*, for details.
Implement the complete security interface in phases	Select the appropriate strategy to implement logon and batch/function security checking in phases.
Troubleshoot security interface errors and understand security interface calls	Identify the type of error and issue the command appropriate for your security software to generate a report regarding the error. See *Troubleshooting Security Interface Errors* on page 306 and *Understanding Security Interface Calls* on page 309.
Implement batch and logon security for SNA and BSC sessions without implementing the Connect:Enterprise security interface	In the ODF, set the parameter SECURITY=. See Chapter 3, *Configuring *OPTIONS Record for System Resources,* for more details.

Following are the requirements for implementing all functions of the security interface:

- The Connect:Enterprise server, all remote sites (SNA and BSC), and all ISPF and CICS users must have a valid user ID defined to their security package.
- All SNA remote sites must supply a password with a remote name at logon.
- All BSC remote sites must supply a valid free-form signon card with a remote name and password.
- All Connect:Enterprise ISPF and CICS users must supply a valid user ID and password at signon.
- Pseudo data set name rules must be in place in the security package for each batch/function access according to mailbox ID and user ID.

The security administrator can perform the tasks required for the first and last items, but the remote sites must perform items 2, 3, and 4. It may take some time for the remote sites to make these changes. *Strategies for Implementing the Security Interface in Phases* on page 303 contains some approaches to phasing in the security interface.

Defining a User/User ID for Connect:Enterprise

Connect:Enterprise must run under a user and user ID with proper authority, whether the security interface is activated or not. This user and user ID must have the authority to:

- Execute as a started task
- Execute in an APF–authorized environment
- Allocate and open the data sets specified in the Connect:Enterprise JCL
- Open the VTAM ACBs used by Connect:Enterprise
- Submit batch jobs to the internal reader
- Dynamically define itself as a console interface. The name used is that specified with the MBXNAME ODF parameter
- Issue the RACROUTE macro to validate user passwords
- Access the OMVS segment (if FTP is enabled)
- Update the SSL key database (if SSL or TLS is enabled)

Note: Connect:Enterprise does not need direct access to the VSAM batch files. The VSAM file server handles all access.

You can use the default user and user ID for Connect:Enterprise execution, but this is not the safest approach. It is better to define a specific user/user ID that you can control and modify without affecting other applications.

Use the following examples to help create a user/user ID for Connect:Enterprise. Text where you must supply site-specific information is shown in lowercase letters. You must have the proper authority to implement these security rules. Also, you may need to alter these rules for your site’s security policies.

Defining a Connect:Enterprise User/User ID in an RACF Implementation

Use the following sample RACF statement to define a Connect:Enterprise user:

```
ADDUSER mboxtsk AUTHORITY(USE) PASSWORD(xxxxx) OWNER(mbx1)
DFLTGRP(starttsk) NAME(’connect:enterprise task’)
```

- MBOXTSK is the user name on which the system checks the authority of the Connect:Enterprise task.

- The OWNER parameter points to a group defined specifically for this Connect:Enterprise system.
- The DFLTGRP parameter points to a group set up for started tasks.

Defining a Connect:Enterprise User/ User ID in a CA-ACF2 Implementation

Use the following sample CA-ACF2 statement to define a Connect:Enterprise user:

```
INSERT mboxtsk STC MAXDAYS(0) NAME(connect:enterprise task)
PASSWORD(zzzzzzzz)
```

MBOXTSK is the user name on which the system checks the authority of the Connect:Enterprise task.

Defining a Connect:Enterprise User ID in a CA–TOP SECRET Implementation

Before creating a Connect:Enterprise user, you must define a facility to keep CA–TOP SECRET from prompting the console for passwords and new passwords when they are not provided by remote users. Create the following facility definition in the CA–TOP SECRET environment (TSS) parameter file. In this example, the facility name is MAILFAC.

```
FAC(USER11=NAME=mailfac)
FAC(mailfac=NOTSOC,RES,NOIJU,AUTHINIT)
```

Use the following TSS statements as an example to define a Connect:Enterprise user:

```
TSS CRE(mboxtsk)TYPE(USER) NAME(connect:enterprise task)+
DEPT(mailbox)FAC(BATCH,STC)PASSWORD(NOPW,0)+
      MASTFAC=(mailfac)
TSS ADD(STC)PROC(mboxtsk)ACID(mboxtsk)
```

In this example, MBOXTSK is the name of the user the system uses to check for authority of the Connect:Enterprise task. The DEPT parameter points to a department Accessor ID (ACID) created specifically for Connect:Enterprise. If you intend to activate the Connect:Enterprise security interface, the value for the DEPT parameter should match the value for the pseudo data set name high-level qualifier (MBXHLQ in the ODF, default is MAILBOX). See *Creating the Security Interface Control Hierarchy* on page 292. The MASTFAC parameter points to the facility created earlier.

Defining the User ID for the VSAM File Server Task

All access to Connect:Enterprise VSAM files is through the VSAM file server task. The Connect:Enterprise server, online users, and offline utilities do not directly access the files. However, unprotected VSAM files can be accessed by programs and users that are not using Connect:Enterprise product.

To protect the VSAM file server task, you must:

- Define the user ID for the VSAM file server task.
- Define data set security rules that protect each VSAM file.

Define the User ID for the VSAM File Server Task in an RACF Implementation

You must define a user ID that associated with the VSAM file server subsystem. Connect:Enterprise VSAM security checking is performed against this user ID.

Use the following RACF statements to implement VSAM file security rules.

The following sample RACF statement defines a user for the VSAM file server:

```
ADDUSER vsmsrv1 AUTHORITY(USE) PASSWORD(xxxxx) OWNER(mbx1)
DFLTGRP(mbx1) NAME(‘vsam server for mbx1’)
```

The following sample RACF statement defines the default data set security rules for the VSAM batch queues:

```
ADDSD ‘your.mailbox.**’ OWNER(vsmsrv1) UACC(NONE)
```

The high-level qualifier of the specified data set name is the name of the clusters defined for Connect:Enterprise.

Define the VSAM File Server Task User ID in a CA-ACF2 Implementation

Use the following sample CA-ACF2 statements to implement VSAM file security rules:

The following CA-ACF2 statement defines a user for the VSAM file server:

```
INSERT vsmsrv1 STC MAXDAYS(0) NAME(vsam file server)  PASSWORD(zzzzzzz)
```

The following CA-ACF2 statement defines the default VSAM batch file data set security rules:

```
your.mailbox.- UID(-) READ(PREVENT) WRITE(PREVENT) ALLOCATE(PREVENT)
```

Define the VSAM File Server Task User ID in a CA–TOP SECRET Implementation

Use the following sample TSS statements to implement VSAM file security rules:

The following TSS statements define a user for the VSAM file server:

```
TSS CRE(vsmsrv01) TYPE(USER) NAME(‘vsam file server mbx1) +
       DEPT(mailbox) FAC(STC) PASSWORD(yyyyyy)
TSS ADD(STC) PROC(vsmsrv1) ACID(vsmsrv01)
```

The following TSS statement defines the default data set security rules for the VSAM batch files:

```
TSS PE(ALL) DSN(your.mailbox.*) ACCESS(NONE)
```

Defining Data Set Security Rules for VSAM Files

In addition to creating a user ID for the VSAM file server task, you must also define data set security rules for the VSAM batch files to restrict access to the VSAM batch files to the following users:

- The VSAM file server
- The Connect:Enterprise system administrator who performs file maintenance
- The DASD administrator who performs file backup and recovery functions

Defining Access to VSAM Files in an RACF Implementation

Use the following sample RACF statements to grant access to the VSAM batch files:

```
PERMIT ‘your.mailbox.**’ ACCESS(UPDATE) ID(vsmsrv1)
PERMIT ‘your.mailbox.**’ ACCESS(ALTER) ID(mbxadmin)
PERMIT ‘your.mailbox.**’ ACCESS(ALTER) ID(dasdadm)
```

Note: The VSAM file server does not require ALTER authority.

Defining Access to VSAM Files in a CA-ACF2 Implementation

Use the following sample CA-ACF2 statements to grant access to the VSAM batch files:

```
your.mailbox.- UID(mbxadmin) READ(ALLOW) WRITE(ALLOW) ALLOCATE(ALLOW)
your.mailbox.- UID(dasdadm) READ(ALLOW) WRITE(ALLOW) ALLOCATE(ALLOW)
```

These examples assume that the user identification string (UID) for each user is the same as the Logon ID. If not, supply the correct UID in each statement.

Note: The VSAM file server does not require ALLOCATE authority.

Defining Access to VSAM Files in a CA–TOP SECRET Implementation

Use the following sample TSS statements to grant access to VSAM batch files:

```
TSS PE(vsmsrv01) DSN(your.mailbox.*) ACCESS(READ,WRITE)
TSS PE(mbxadmin) DSN(your.mailbox.*) ACCESS(ALL)
TSS PE(dasdadm) DSN(your.mailbox.*) ACCESS(ALL)
```

Note: The VSAM file server does not require SCRATCH or CREATE access.

Controlling Remote Access with the Logon Security Interface

The optional logon security interface controls access to Connect:Enterprise sessions initiated through remote connects or APPC. The interface uses System Authorization Facility (SAF) calls to your security package to validate a user-supplied password. To use the logon security interface, the Connect:Enterprise load library must be an APF- authorized library and the Connect:Enterprise load module, STMAIN, must be linked AC=1.

To implement the logon security interface, you must define a user ID (the remote name) and password within your security package for each remote site that accesses Connect:Enterprise. See *Creating the Security Interface Control Hierarchy* on page 292 for sample definitions for each security package. Each remote site must then supply its remote name and password to identify itself when initiating a session with Connect:Enterprise. See *BSC Logon Format* on page 287 to see how to format this information.

After the remote site supplies the logon information, it passes to the security interface. (If you use security exit one, the exit validates the information and accepts the logon before passing it on to the security interface.) The security interface then uses SAF to validate the remote name and password with your system security package. Connect:Enterprise does not store or maintain the password.

If the SAF call is successful, the remote continues the session with Connect:Enterprise. If the SAF call fails, the session terminates and SAF sends a message to the remote and to the for z/OS console. The remote connect summary log record indicates that the session failed due to password validation errors.

If the remote site supplies an optional new password, an SAF request to change the password is included in the initial password check. The remote password changes to the new password after the initial password check completes successfully.

The security interface processes logon requests from Connect:Enterprise ISPF and CICS interfaces and CICS API users in the same way. A user enters a user ID and password at logon. The system passes this information to the APPC security exit, if used, for validation and default processing before calling the logon security interface.

Note: The logon security check does not require the security exit one and APPC security exits.

To revoke a remote user’s access or change a user’s password, use your security package’s administrative interface. No changes are required within Connect:Enterprise.

BSC Logon Format

The remote user must supply the remote name and password fields in a free-form SIGNON card as the first record sent to Connect:Enterprise. The *SIGNON section of the ODF defines the format for the *SIGNON card.

For more information on the free-form BSC SIGNON card format, see *Free-Form BSC SIGNON* on page 175.

Note: Connect:Enterprise only supports a single password for the remote. It does not support the BSC line password.

SNA Logon Format

The remote user must supply the remote name and passwords in a valid LOGON command. The format for this command is:

```
LOGON APPLID(xxxxxxxx) LOGMODE(xxxxxxxx) DATA(remotenm,,password,newpass)
```

Parameter	Description
remotenm	The remote name. This parameter is required. It has a maximum length of 8 characters and must be uppercase text.
password	The logon password. This parameter is required. The security system defines this parameter’s length and syntax. A password is usually a maximum of 8 characters and must be uppercase text. It is a positional parameter and it must be preceded by two commas. **Note:** Connect:Enterprise only supports a single password for the remote. It does not support the SNA line password.
newpass	The new logon password. This is an optional parameter. The security system defines the length and syntax of this parameter. The remote user only specifies this parameter when changing a password. If this parameter is omitted, the preceding comma is not required. This must be in uppercase text.

FTP Logon Format

The remote user must supply the remote name and passwords in valid USER and PASS commands.

The format for the USER command is:

```
USER remotenm
```

The parameter for the USER command is:

Parameter	Description
remotenm	The remote name. This parameter is required. It has a maximum length of 8 characters and must be uppercase text.

The format for the PASS command is:

```
PASS password[/newpass/newpass]
```

The parameters for the PASS command are:

Parameter	Description
password	The logon password. This parameter is required. The security system defines this parameter's length and syntax. A password is usually a maximum of 8 characters and must be uppercase text. It is a positional parameter and it must be preceded by a slash.
newpass	The new logon password. This is an optional parameter. The security system defines the length and syntax of this parameter. The remote user only specifies this parameter when changing a password. If this parameter is omitted, the preceding slash is not required.

Controlling User Processing of Batches with the Batch/Function Security Interface

The batch/function security interface controls the functions invoked by remote, APPC, and offline utility users to process specific batches. The batch/function security interface uses SAF calls to your system security package to perform security checks. You can restrict access by Connect:Enterprise system, function, Mailbox ID, and logon ID through data set name rules defined in the security package.

The security interface uses SAF to make the following batch or function checks:

- When processing a $$ online command, the security interface verifies that the user (remote name) has permission to execute the specified function on the specified batch for the specific Connect:Enterprise system.
- When processing an offline utility command, the security interface verifies that the user (user ID) has permission to execute the specified function on a specific batch for the specified Connect:Enterprise system.
- When making requests to Connect:Enterprise through the CICS interface, the ISPF interface, or the CICS API, Connect:Enterprise verifies that the user provided in the IPS has permission to execute the request.

To use the batch/function security interface, the Connect:Enterprise load library must be an APF-authorized library, the Connect:Enterprise load module STMAIN must be linked AC=1, and the Connect:Enterprise load module STOUTL must run from an APF -authorized load library and be linked AC=1.

Batch/Function Security Data Set Verification

Connect:Enterprise batch or function security verifies that the remote, APPC user, or STOUTL job has UPDATE authority for the generated data set name used for the Connect:Enterprise system and batch/function being processed. (This is a pseudo data set name; the data set is not physically created.)

You can restrict access by remote sites, APPC users, or batch jobs to this data set name by defining data set access rules in your security package.

The data set name has the following format:

```
<MBXHLQ>.<MBXNAME>.<Subsystem>.<Function>.<ID>
```

It contains the following nodes:

Node	Description
<MBXHLQ>	The Connect:Enterprise pseudo data set name high-level qualifier specified in the ODF *OPTIONS parameter MBXHLQ. The default is MAILBOX.
<MBXNAME>	The Connect:Enterprise system name specified in the ODF *OPTIONS parameter MBXNAME. The default is MAILBOX.
<Subsystem>	The subsystem issuing the check: ONLINE, APPC, or OFFLINE. This value is system-generated.
<Function>	The function being validated, for example, $$ADD, C$A20, or ADD. This value is system-generated. FTP commands are mapped as follows: **FTP Command** **Function** STOR/STOU $$ADD LIST/NLST $$DIR RETR $$REQ DELE $$DEL
<ID>	The Mailbox ID of the batch specified in the request. This value is system-generated.

The exact data set name nodes used can vary depending on the type of access (online command, offline utility, or APPC). The following sections describe the data name nodes by type of access.

Batch/Function Security Checks for Online Command Processing

For Connect:Enterprise online command processing checks, the generated data set name uses the following format:

```
<MBXHLQ>.<MBXNAME>.ONLINE.<Function>.<ID>
```

For example, if a remote site performs a $$REQ command for a batch named FINANCED on a Connect:Enterprise system where MBXHLQ=MAILBOX and MBXNAME=MBX1, the batch/function security checks for UPDATE authority to MAILBOX.MBX1.ONLINE.$$REQ.FINANCED.

Batch/Function Security Checks for Offline Utility Processing

For offline utility processing checks, the generated data set name uses the following format:

```
<MBXHLQ>.<MBXNAME>.OFFLINE.<Function>.<ID>
```

The last node (ID) is not used for the PURGE function.

For example, if a user requests the offline utility ERASE for batch ALTOONA on a Connect:Enterprise system with MBXHLQ=MBXSUP and MBXNAME=MBX2, the batch/function security checks for UPDATE authority for data set name MBXSUP.MBX2.OFFLINE.ERASE.ALTOONA.

Batch/Function Security Checks for APPC Processing

For Connect:Enterprise APPC processing checks, the generated pseudo data set name consists of the following four or five nodes:

```
<MBXHLQ>.<MBXNAME>.APPC.<IPS request ID>.<ID>
```

The last node <ID> is not generated unless the IPS request ID is C$U22 (Option 2.2.1.2, Batch Browse from CICS interface) or C$U31 (Option 22.1, Batch Browse from ISPF interface). In these cases, this node contains the batch ID of the requested batch.

For example, if an APPC user submits a $$CONNECT request through the ISPF interface on a Connect:Enterprise system with MBXHLQ=MBOX and MBXNAME=MBX2, the batch/function security would check for UPDATE authority for data set name MBOX.MBX2.APPC.C$O03.

For a summary of all APPC request types, see the *Connect:Enterprise for z/OS Application Agent and User Exits Guide*. Complete details of each request are documented in the IPS mapping macros included in the CE.SOURCE file on the installation tape.

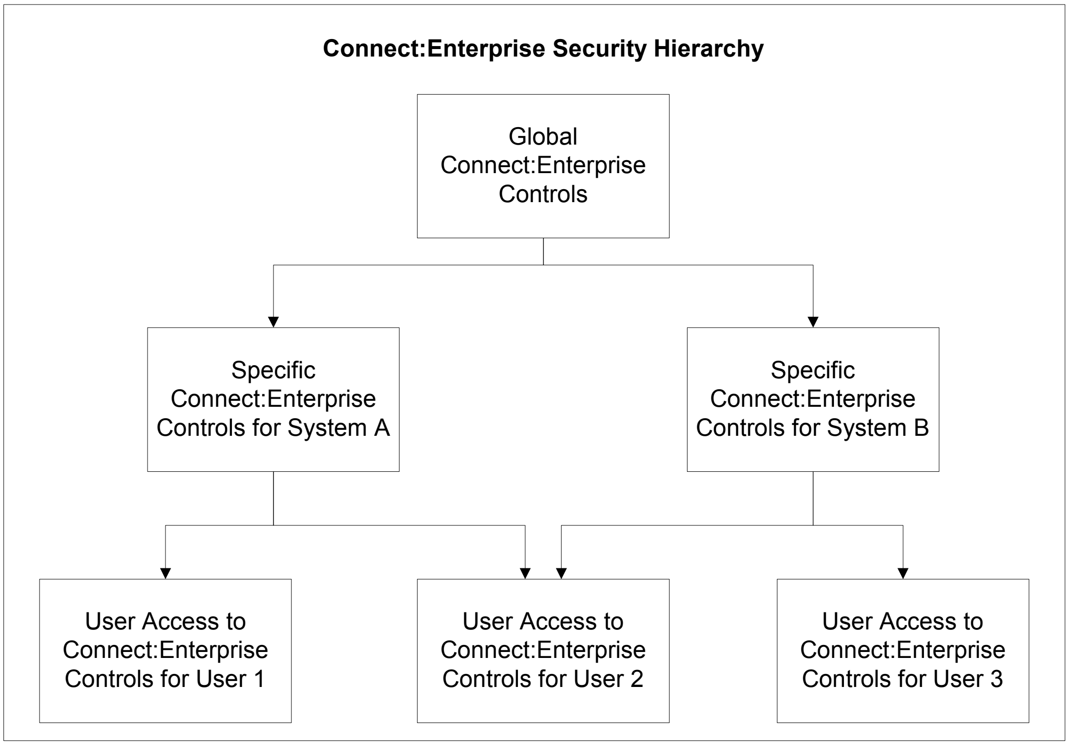
Note: When logging on to a Connect:Enterprise through the ISPF and CICS interfaces, the interfaces send a VERIFY request. No IPS is supplied. This request uses the logon security function, not the batch/function.

Security Checks within the VSAM File Server

Automatic security checks within the VSAM file server determine whether or not the server has access to a data set before processing an ALLOCATE request. These checks prevent S913 ABENDs when security access is not allowed.

Creating the Security Interface Control Hierarchy

To implement the security interface, you create a hierarchy of controls to Connect:Enterprise logon and batch/function security. The first level of control contains global security rules for all Connect:Enterprise systems. The second level of control is for an individual Connect:Enterprise system. The third level of control is for individual users of each Connect:Enterprise system.


Connect:Enterprise Security Hierarchy
Global Connect:Enterprise Controls
Specific Connect:Enterprise Controls for System A
Specific Connect:Enterprise Controls for System B
User Access to Connect:Enterprise Controls for User 1
User Access to Connect:Enterprise Controls for User 2
User Access to Connect:Enterprise Controls for User 3

Choose a Global Name for All Connect:Enterprise Security Checks

To build a proper control hierarchy for all Connect:Enterprise systems, the security administrator must define a high-level qualifier that fits the site's security standards. This name is used as an anchor for all security rules for all Connect:Enterprise systems. Also, Connect:Enterprise uses this name in batch/function security as the high-level qualifier for the pseudo data set name.

Use the ODF *OPTIONS parameter or the PURGE offline utility parameter MBXHLQ to supply a high-level qualifier that meets your site's security naming standards. Use this name for all your Connect:Enterprise systems (production A, production B, test), because the second node of the pseudo data set name is the MBXNAME value, which specifies a name for specific Connect:Enterprise systems.

If you do not supply a global name in the ODF, the default is MAILBOX.

Name Each Connect:Enterprise System

To implement any portion of the Connect:Enterprise security interface, define a unique name for each Connect:Enterprise system. This name is used for:

- Logon security checks
- Batch/function security checks for remote and local users
- Connect:Enterprise system identification to the ISPF and CICS interfaces

Use the ODF *OPTIONS parameter or the PURGE offline utility parameter MBXNAME to supply this system's name. If you do not supply the name, the default is MAILBOX.

Connect:Enterprise stores the MBXNAME value in the VPF so that the offline utilities can use it. It updates the value in the VPF each time Connect:Enterprise starts.

Note: You cannot change the MBXNAME value with the ISPF/CICS interfaces.

The name you specify in the MBXNAME parameter should match the name that identifies a Connect:Enterprise system in the ISPF and CICS interfaces.

Define Security Rules

To implement logon security controls and batch/functions controls, create the following:

- One or more security definitions that identify all Connect:Enterprise systems as an entity
- One or more security definitions that identify each Connect:Enterprise system
- One or more security definitions that identify each Connect:Enterprise user with a logon password
- One or more security definitions that link each user to one or more specific Connect:Enterprise systems
- One or more security rules that protect batches/functions at a global (default) level
- One or more security rules that grant access to batches/functions for specific users of specific Connect:Enterprise systems

You can define these controls several ways, usually based on the global security rules previously defined for your for z/OS system. Refer to the procedures in *Creating the Control Hierarchy in an RACF Environment* on page 293, *Creating the Control Hierarchy in a CA-ACF2 Environment* on page 297, or *Creating the Control Hierarchy in a CA–TOP SECRET Environment* on page 299 for instructions on creating the control hierarchy in the different environments.

Creating the Control Hierarchy in an RACF Environment

Perform the following steps to implement the control hierarchy in a RACF environment:

1. Define a group profile for global Connect:Enterprise system controls.
2. Define a group profile and user/user ID for each Connect:Enterprise system.

3. Define each Connect:Enterprise user and identify them to their default Connect:Enterprise system.
4. Define a connection to any other Connect:Enterprise system for the user.
5. Define default batch/function security rules for each Connect:Enterprise system.
6. Define batch/function security rules for users that require a different security level from the default batch/function security rules.

Each of these steps is described in detail in the following sections.

To implement these security rules, you must have the proper RACF authority. Contact your system security administrator for assistance.

Alter these security rules as necessary to comply with your security policies.

Note: The following rules assume *enhanced generic naming* is active.

Define a Group Profile for Global Connect:Enterprise System Controls

To define a group profile for global Connect:Enterprise system controls, use the following RACF statement:

```
ADDGROUP MAILBOX OWNER(SYS1) SUPGROUP(SYS1)
```

MAILBOX is the name of the defined group. This name should match the pseudo data set name high-level qualifier (MBXHLQ) name.

Use the following RACF statement to define the default security rule for all data set names (batch/function) that are checked with MAILBOX as the high-level qualifier:

```
ADDSD ‘MAILBOX.**’ UACC(NONE)
```

Use this rule for all Connect:Enterprise systems that do not have a more specific security rule. Substitute the name used in the previous item for MAILBOX, if different.

Define a Group Profile and User/User ID for Each Connect:Enterprise System

Perform the following procedure to define a group profile for each Connect:Enterprise system. Repeat this procedure for each Connect:Enterprise system that security interface will protect.

Use the following RACF statement to define a group profile for each Connect:Enterprise:

```
ADDGROUP MBX1 OWNER(MAILBOX) SUPGROUP(MAILBOX)
```

In this example,

- MBX1 is the name of the defined group. It is also the name specified in the *OPTIONS parameter MBXNAME for the specific Connect:Enterprise system.
- MAILBOX is the name chosen for the higher level group in the previous step.

Use the following RACF statement to define the default security rule for all pseudo data set names (batch/function) that are checked with MBX1 as the Connect:Enterprise name and MAILBOX as the high-level qualifier. Substitute the name used in the previous step for MAILBOX, if different.

```
ADDSD 'MAILBOX.MBX1.**' UACC(NONE)
```

Each Connect:Enterprise system also needs its own user/user ID to properly function. If not already done, follow the steps in *Defining a User/User ID for Connect:Enterprise* on page 283 to set up a user ID for each Connect:Enterprise system.

Define Each Connect:Enterprise User to a Default Connect:Enterprise System

This step is only required to define remote Connect:Enterprise users. It is not necessary for offline utility users, because they already have defined user IDs.

Use the following RACF statement to define a Connect:Enterprise remote user and identify the user to the default Connect:Enterprise system:

```
ADDUSER REMOTE1 AUTHORITY(USE) PASSWORD(XXXXX) DFLTGRP(MBX1) NAME('REMOTE USER 1')
```

In this example:

- REMOTE1 is the name of the remote user defined to Connect:Enterprise.
- MBX1 is the name of the default Connect:Enterprise system to which REMOTE1 connects.
- The PASSWORD parameter specifies a temporary password (XXXXX) for the first security check. After the first check, RACF requires the user to enter a new password. If you do not want to make the user change a password at first login, you can logon to TSO as the user, change the password yourself, and provide the user with the new password (assuming that you have authority to do so).

You are also prompted to change passwords on a system-specific interval. To keep the password from regularly expiring, use the following RACF statement:

```
PASSWORD NOINTERVAL USER(REMOTE1)
```

Define a Connection to Another Connect:Enterprise System for the User

You need to define additional connections for users who use offline utilities and did not have a Connect:Enterprise user ID created during the previous step. This step is also required if a remote user connects to more than one Connect:Enterprise system.

Use the following RACF statement to define additional connections for the remote user to other Connect:Enterprise systems:

```
CONNECT REMOTE1 GROUP(MBX2) OWNER(MBX2)
```

REMOTE1 is the name of the remote user defined to Connect:Enterprise. MBX2 is the name of the additional Connect:Enterprise system to which REMOTE1 can connect.

Note: You must define the MBX2 group profile to RACF before connecting users to it.

Define Default Batch/Function Security Rules

Use the following sample RACF statements to set default security rules for batches/functions within a specific Connect:Enterprise system. In the following examples the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statement sets a default security rule allowing all remote users to use the $$DIR command on system MBX1:

```
ADDSD ‘MAILBOX.MBX1.ONLINE.$$DIR.**’ UACC(UPDATE) OWNER(MBX1) NOSET GENERIC
```

The following statement sets a default security rule allowing all users to add batches offline to system MBX1:

```
ADDSD ‘MAILBOX.MBX1.OFFLINE.ADD.**’ UACC(UPDATE) OWNER(MBX1) NOSET GENERIC
```

The following statement sets a default security rule allowing all ISPF and CICS interface users to communicate with MBX1 and to use all ISPF and CICS functions:

```
ADDSD ‘MAILBOX.MBX1.APPC.**’ UACC(UPDATE) OWNER(MBX1) NOSET GENERIC
```

Define Exception Batch/Function Security Rules for Users

Use the following RACF statement examples to create user-specific security rules for a specific Connect:Enterprise system. In the following examples, the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statement enables the REMOTE1 and REMOTE2 users to use the EXTRACT offline utility to extract batches beginning with Mailbox ID “ID37”:

```
PERMIT ‘MAILBOX.MBX1.OFFLINE.EXTRACT.ID37*’ ACCESS(UPDATE) ID(REMOTE1 REMOTE2)
```

The following statement prevents user REMOTE2 from issuing a $$SHUTDOWN request from the ISPF or CICS interfaces:

```
PERMIT ‘MAILBOX.MBX1.APPC.C$O16’ ACCESS(NONE) ID(REMOTE2)
```

The following statement enables user REMOTE1 to use the $$DEL remote command on batches beginning with Mailbox ID “ID37”:

```
PERMIT ‘MAILBOX.MBX1.ONLINE.$$DEL.ID37*’ ACCESS(UPDATE) ID(REMOTE1)
```

Creating the Control Hierarchy in a CA-ACF2 Environment

Perform the following steps to implement a control hierarchy in a CA-ACF2 environment:

1. Define global default batch/function security rules for all Connect:Enterprise systems.
2. Define a logon ID (LID) for each remote Connect:Enterprise user.
3. Define default batch/function security rules for each Connect:Enterprise system.
4. Define batch/function security rules for users that require a security level different from the default batch/function security rules.

Each of these steps is described in detail in the following sections.

To implement these security rules, you must have the proper CA-ACF2 authority. Contact your system security administrator for assistance.

Alter these security rules as necessary to comply with your security policies.

Note: After you create new security rules or change existing ones, you must compile and store them in the CA-ACF2 rule database. See the *ACF2 Administrator Guide* for more information.

Define Global Default Batch/Function Security Rules for All Connect:Enterprise Systems

Use the following sample CA-ACF2 statement to set a default security rule that prevents all users from accessing any batch/function in any Connect:Enterprise system.

```
$KEY(MAILBOX)
$MODE(ABORT)
$OWNER(Connect:ENTERPRISE)
- UID(-) WRITE(PREVENT)
```

The $KEY statement value should match the pseudo data set name high-level qualifier (MBXHLQ) value.

Define a Logon ID (LID) for Each Remote Connect:Enterprise User

This step is only required to define remote users of Connect:Enterprise. This step is not necessary for offline utility users because they have defined user IDs.

1. Use the following CA-ACF2 statement to define a remote Connect:Enterprise user:

   ```
   INSERT REMOTE1 MAXDAYS(0) NAME(REMOTE USER 1) PASSWORD(XXXXX)
   ```

   In this example:

   - REMOTE1 is the name of the remote user you want to define to Connect:Enterprise.
   - MAXDAYS(0) specifies that CA-ACF2 does not automatically require users to change their password.
   - The PASSWORD parameter specifies a temporary password (XXXXX) for the first security check. After the first check, CA-ACF2 requires the user to enter a new password.

If you do not want to make the user change a password at first login, logon to TSO as the user, change the password yourself, and provide the user with the new password (assuming that you have authority to do so).

2. Before proceeding with the remaining steps, you must know the UID assigned to each user by CA-ACF2.

Define Default Batch/Function Security Rules for Each Connect:Enterprise System

Use the following sample CA-ACF2 statements to set default security rules for batches/functions within a specific Connect:Enterprise system.

In the following examples the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statement sets a default security rule that prevents all users from accessing any batch/function in Connect:Enterprise system MBX1:

```
MAILBOX.MBX1.- UID(-) WRITE(PREVENT)
```

The following statement sets a default security rule that enables all remote users to use the $$DIR command on Connect:Enterprise system MBX1:

```
MAILBOX.MBX1.ONLINE.$$DIR.- UID(-) WRITE(ALLOW)
```

The following statement sets a default security rule that enables all users to offline add batches to Connect:Enterprise system MBX1:

```
MAILBOX.MBX1.OFFLINE.ADD.- UID(-) WRITE(ALLOW)
```

The following statement sets a default security rule that enables all ISPF and CICS interface users to communicate with Connect:Enterprise MBX1, and use all ISPF and CICS functions:

```
MAILBOX MBX1.APPC.- UID(-) WRITE(ALLOW)
```

Define Batch/Function Security Rules for Users

Use the following sample CA-ACF2 statements to create user–specific security rules for a specific Connect:Enterprise system.

In the following examples the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statement enables users with an XXXXXXX UID to use the EXTRACT utility to extract batches whose Mailbox ID begins with ID37 from Connect:Enterprise system MBX1:

```
MAILBOX.MBX1.OFFLINE.EXTRACT.ID37- UID(XXXXXXX) WRITE(ALLOW)
```

The following statement prevents users with a YYYYYYY UID from issuing a $$SHUTDOWN request to Connect:Enterprise system MBX1, from the ISPF or CICS interfaces:

```
MAILBOX.MBX1.APPC.C$O16- UID(YYYYYYY) WRITE(PREVENT)
```

The following statement enables all users with a ZZZZZZZ UID to use the $$DEL command for batches whose Mailbox ID begins with ID37 on Connect:Enterprise system MBX1:

```
MAILBOX.MBX1.ONLINE.$$DEL.ID37- UID(ZZZZZZZ) WRITE(ALLOW)
```

Creating the Control Hierarchy in a CA–TOP SECRET Environment

The following steps allow you to implement a hierarchy of control in a TSS environment:

1. Define a Department Accessor ID (ACID) for all Connect:Enterprise users.
2. Define the High-level Qualifier (HLQ) for Connect:Enterprise systems.
3. Define a User Accessor ID (ACID) for each Connect:Enterprise system.
4. Define a User Accessor ID (ACID) for each user of Connect:Enterprise
5. Define default batch/function security rules for each Connect:Enterprise system.
6. Define batch/function security rules for users that require security access differing from that provided by the default batch/function security rules.

Each of these steps is described in detail in the following sections.

You must have the proper TSS authority to implement the following security rules. Contact your system security administrator for assistance.

Define a Department Accessor ID (ACID) for Connect:Enterprise Users

Use the following TSS command to define an ACID for Connect:Enterprise users that do not have a TSS ACID:

```
TSS CRE(MAILBOX) TYPE(DEPT) NAME('Connect:Enterprise') DIV(ownerdiv)
```

In this example:

- TYPE(DEPT) is a high-level group definition.
- MAILBOX is the ACID of the department defined for Connect:Enterprise. The CRE value should match the pseudo data set name high-level qualifier (MBXHLQ) value.
- NAME is a description of the department.

Define the High-level Qualifier for all Connect:Enterprise Systems

Use the following TSS command to define and secure the MAILBOX high-level qualifier:

```
TSS ADD(MAILBOX) DSN(MAILBOX.*)
```

The ADD and DSN values should match the value used in the pseudo data set name high-level qualifier (MBXHLQ).

Define a User ACID for Each Connect:Enterprise System

Follow the steps in *Defining a User/User ID for Connect:Enterprise* on page 283 to set up a user ID for each Connect:Enterprise system.

Define a User Accessor ID for Each User

This step is only required to define remote users of Connect:Enterprise.

Use the following TSS command to define ACIDs for each remote Connect:Enterprise user:

```
TSS CRE(REMOTE1) TYPE(USER) NAME(‘REMOTE USER 1’) PASSWORD(yyyyy,0)
```

In this example:

- REMOTE1 is the ACID of the remote user defined to Connect:Enterprise.
- PASSWORD(yyyyy,0) indicates that the user does not have to change the password at the interval specified in the TSSPARM0 member.
- A profile is added to the ACID that establishes which Connect:Enterprise systems the ACID can use.

Define Default Batch/Function Security Rules for Each Connect:Enterprise System

Use the following sample TSS commands to set default security rules that permit all remote users to use specific functions on specific Connect:Enterprise systems.

In the following examples the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statement sets a default security rule that enables all remote users to use the $$DIR command on Connect:Enterprise system MBX1:

```
TSS PE(ALL) DSN(MAILBOX.MBX1.ONLINE.$$DIR.*) ACCESS(UPDATE)
```

The following statement sets a default security rule to allow all users to offline add batches to Connect:Enterprise system MBX1:

```
TSS PE(ALL) DSN(MAILBOX.MBX1.OFFLINE.ADD.*) ACCESS(UPDATE)
```

The following statement sets a default security rule that enables all ISPF and CICS interface users to communicate with Connect:Enterprise system MBX1, and to use all ISPF/CICS functions:

```
TSS PE(ALL) DSN(MAILBOX.MBX1.APPC.*) ACCESS(UPDATE)
```

These statements assume that the TSS ALL record is utilized. If the ALL record is not utilized, create PROFILE type ACIDs. You can then define permissions for the preceding data sets for the PROFILE ACIDs. See your security administrator to determine if this is required.

Define Exception Batch/Function Security Rules

Use the following sample TSS commands to define user-specific Connect:Enterprise system access.

In the following examples the global Connect:Enterprise system name is MAILBOX. The specific system name is MBX1.

The following statements allow the REMOTE1 and REMOTE2 users to use the EXTRACT utility to extract batches whose Mailbox ID begins with ID37 from Connect:Enterprise system MBX1.

```
TSS PE(REMOTE1) DSN(MAILBOX.MBX1.OFFLINE.EXTRACT.ID37*) ACCESS(UPDATE)
TSS PE(REMOTE2) DSN(MAILBOX.MBX1.OFFLINE.EXTRACT.ID37*) ACCESS(UPDATE)
```

Note: You can also accomplish this by creating a profile, giving the profile the permissions described in *Define Default Batch/Function Security Rules* on page 296, then adding the profile to each user.

The following statement enables the REMOTE1 user to use the $$DEL remote command on batches whose Mailbox ID begins with ID37 on Connect:Enterprise system MBX1:

```
TSS PE(REMOTE1) DSN(MAILBOX.MBX1.ONLINE.$$DEL.ID37*) ACCESS(UPDATE)
```

The following statement prevents user REMOTE2 from issuing the $$SHUTDOWN request from the ISPF or CICS interface on Connect:Enterprise system MBX1:

```
TSS PE(REMOTE2) DSN(MAILBOX.MBX1.APPC.C$O16*) ACCESS(NONE)
```

Activating the Security Interface

You can activate or inactivate security on one or more of the following levels:

- The default is no level at all, meaning there is no security checking. The OFF value for the MBXSECURE ODF *OPTIONS parameter is automatically set to inactivate security interface checking at the global level.
- A system-wide level so that security checking is performed for all protocols. See the values for the MBXSECURE ODF *OPTIONS parameter below.

- A protocol-specific level, for example, for FTP connections only. Parameters are provided for the following protocols: Binary Synchronous (BSC) connections, Systems Network Architecture (SNA) connections, FTP connections, APPC LU6.2 connections, InterConnect Option APPC LU6.2 connections, Cross System Client APPC LU6.2 connections, CICS and ISPF User Interface APPC LU6.2 connections, and STOUTL offline utility functions. See the parameter for the specific type of connection you want to control.

Security can be implemented for logon checking only (LOGON), batch/function checking only (BATCH), or both (ALL). These three values are provided for all parameters. In addition, you can use the WARN value to activate security checking without interrupting processing while displaying an error message indicating that security has been violated. The WARN value is available for all global and protocol-specific parameters. You can activate security checking at a system-wide level and inactivate it for a specific protocol by turning it off for that type of connection. See *Chapter 3, Configuring *OPTIONS Record for System Resources,* for detailed information on the security parameters and their values.

*Sample Security ODF *OPTIONS Records* on page 302 illustrates different security settings. If you do not specify a value for a particular protocol parameter, that parameter inherits the same value in effect for MBXSECURE. If you specify a value for a particular protocol parameter, that parameter value overrides the MBXSECURE setting.

You must specify your security preferences in the *OPTIONS record of the Options Definitions File – you cannot change it through the ISPF or CICS interfaces. However, you can view the values set for these security-related parameters using the *OPTIONS Record Parameter Display (Part 1 of 3) screen of the ISPF or CICS interface.

Note: For all transactions that do not originate from a Connect:Enterprise for z/OS product component, for example, for Gentran and user-written API programs, valid user and password values must be set in the H00SUSER and H00PSWD header fields to pass logon security checking.

The offline utilities security interface is automatically activated when online Connect:Enterprise is started with a valid MBXSECURE (BATCH, ALL or WARN) or STLSECURE (BATCH or WARN) value. Offline utility batch/function checking functions the same as online Connect:Enterprise batch/function checking. Offline utilities do not use logon security checking.

The VSAM file server security interface is automatically activated when the server starts.

Sample Security ODF *OPTIONS Records

This section provides examples of the *ODF OPTIONS record with different security configurations.

Note: If you do not specify a value for MBXSECURE or any of the other security-related parameters, no security checking is performed.

The following example specifies to perform the following security checking:

- Batch/function checking for bisync connections.
- Both logon and batch/function checking for FTP connections.

- No security checking for SNA connections.
- Both logon and batch/function checking for API connections where processing will not be interrupted but a warning message will display when a security violation occurs.

```
*OPTIONS
  BSCSECURE=BATCH
  FTPSECURE=ALL
  SNASECURE=OFF
  APISECURE=WARN
```

The next example specifies to perform the following security checking:

- Both logon and batch/function checking for FTP connections.
- Both logon and batch/function checking for ISPF and CICS User Interface APPC LU6.2 connections.
- Logon checking for Cross System Client APPC LU6.2 connections.
- Logon checking for InterConnect Option APPC LU6.2 connections.
- Both logon and batch/function checking for API connections where processing will not be interrupted but a warning message will display when a security violation occurs.

```
*OPTIONS
  FTPSECURE=ALL
  UIFSECURE=ALL
  CSCSECURE=LOGON
  ICOSECURE=LOGON
  APISECURE=WARN
```

Strategies for Implementing the Security Interface in Phases

Because of the time and effort involved to fully implement the security interface, you may want to implement it in phases. Select the strategy for implementing security functions that best meets the requirements of your environment.

Implementing Phased Logon Security Checking

Implementing the security interface in phases provides remote sites time to add SIGNON cards (BSC) or include passwords on their LOGON statements (SNA). For a phased implementation, use one or more of the following strategies:

- Specify the WARN value for the desired security-related parameter, such as MBXSECURE or FTPSECURE, in the ODF *OPTIONS record to allow online logon security checking where a violation does not stop processing.
- Specify the BATCH value for the desired security-related parameter, such as UIFSECURE, in the ODF *OPTIONS record to bypass logon password checking. Security error messages do not display when the user logs on to Connect:Enterprise.

- Code security exit one to set the logon security check complete bit (XX$LGNOK) on in the X1$SFLAG field to bypass logon password checking. This bit can be set selectively or always based on the remote name, remote type, or both. When this bit is on, security exit one indicates that the security interface should not check logon security because it has completed. Security error messages do not display when the user logs on to Connect:Enterprise.
- Code security exit one to supply a password for an SNA remote site that does not include one in its logon USERDATA field. This keeps the security interface from rejecting a logon without a password.
- Code security exit one to supply a remote name for a BSC remote that fails to send a free-form SIGNON card.

  Set the X1$RMTNM to a valid user ID and set the logon security check complete bit (XX$LGNOK) on in the X1$SFLAG field. To help identify the remote, the exit is supplied with the BSC line ID and type, $$command type, Mailbox ID (if supplied on a $$ command), user batch ID (if supplied), password value (if supplied), and a pointer to the input block being processed. If the remote cannot be adequately identified, the exit can set a default user ID for the remote name. For SNA remote sites, the exit should never change the X1$RMTNM field - it must match a *REMOTE entry.

  If the logon security check complete bit indicates a successful logon, the security interface bypasses the logon security check. If the security interface detects that logon processing has not completed, it attempts to do the logon security check prior to the batch/function security check. If the logon security check fails, the session terminates. If the logon security check is successful (or bypassed), the batch/function security check executes.
- Code security exit two to set the X2$ACODE field value to 04 for all security violations. You can set this value selectively or always based on the remote name, remote type, or both, and the security error type. When this value is set, security exit two indicates that the security interface attempts to check logon security. However, security exit two does not stop any processing if it fails. All security errors are written to the console log and can be reviewed without checking the security package log data.

Note: Security exit two is called if either the security interface or security exit one detects an error.

Implementing Phased Batch/Function Security Checking

For a phased implementation of batch/function checking for online Connect:Enterprise, use one or more of the following strategies:

- Specify the WARN value for the desired security-related parameter, such as MBXSECURE or FTPSECURE, in the ODF *OPTIONS record to allow batch/function security checking where a violation does not stop processing.
- Specify the LOGON value for the desired security-related parameter, such as UIFSECURE, in the ODF *OPTIONS record for logon checking only. When the user continues processing batches, no security error messages are displayed.
- Code security exit one to set the batch/function security check complete bit (XX$BCHOK) on in the X1$SFLAG field to use the security interface for logon checking only. This bit can be set selectively or always based on the remote name, remote type, or both. When this bit is on, security exit one indicates that the security interface should not check batch/function security

because it has completed. When the user continues processing batches, no security error messages are displayed.

- For BSC remote sites, code security exit one to set the X1$RMTNM field to a default user ID if the remote site fails to send one.

Note: Never change the X1$RMTNM field for SNA remote sites—this field must match a *REMOTE entry.

- Code security exit two to set the X2$ACODE value to 04 for all security violations. This value can be set selectively or always based on the remote name, remote type, or both, and the security error type. When this value is set, security exit two indicates that the security interface attempts to check logon security. However, security exit two does not stop any processing if it fails. All security errors are written to the console log and can be reviewed without checking the security package log data.

The following sections describe some considerations when using security exits for security checking.

User Exits and the Security Interface

To implement the security interface when you already use security exits one and two for BSC and/or SNA security checking, use one or more of the following strategies:

- If security exit one is the only control over remote security checking, always set the logon security check complete bit (XX$LGNOK) on and the batch/function security check complete bit (XX$BCHOK) on in the X1$SFLAG field. This prevents the security interface from making any security checks for remote sites. The same results can occur if you do not set MBXSECURE.
- Specify MBXSECURE=LOGON or MBXESCURE=ALL (depending on the security level desired for ISPF and CICS users) in the ODF and use security exit one to control access by ISPF and CICS users, and let security exits one and two continue to control online remote accesses.

Note: The BSCSECURE= and/or SNASECURE= parameters may be used to override the global security interface parameter, MBXSECURE=.

- If the user exit wants the security interface to execute override remote logon checking when user exit authorization fails, set the logon security check complete bit (XX$LGNOK) off and exit with a return code of zero. The security interface can execute its logon security check.
- If the user exit wants the security interface to execute override batch/function checking when user exit authorization fails, set the batch/function security check complete bit (XX$BCHOK) off and exit with a return code of zero (even when authorization is not allowed). The security interface can execute its batch/function security check and override the first check.
- If the user exit wants the security interface to execute override batch/function checking when user exit authorization fails, set the logon security check complete bit (XX$LGNOK) off and exit with a return code of zero (even when authorization is not allowed). The security interface can execute its logon security check and override the first check.

User Exits and Security Interface Sequence

The sequence between user exits and the security interface is:

- User exits are called *before* the security interface.
- The security interface is only called if:
  - The security interface is active through one or more security-related parameters in the ODF *OPTIONS record.
  - User exits return a value of zero as a return code. If the user exit does not exist, a return code of zero is assumed.
  - The appropriate logon security checking and batch/function complete bits are off.

Resource Control

Do not use both the user exits and the security interface to control the same resources. For example, use the security interface for logon checking and use the user exit for batch/function checking, because each is a separate resource. Do not use both a user exit and the security interface for batch/function checking. Permanently using both security tools for control of the same resource creates a security exposure. Consider the security exposure risk if you implement the security interface in phases.

Use similar strategies to control authority to APPC functions and offline utilities. Offline utilities do not check for logon authority.

Troubleshooting Security Interface Errors

If problems arise during security interface implementation, use the features in your existing security package to determine the cause of the error. Refer to the documentation with your security package for procedures and appropriate resolutions for security problems.

The following sections provide general information to identify the cause of a Connect:Enterprise security problem.

Violation Reports

Each security package provides security violation reports. The following types of violation reports can help you determine what caused a security interface error:

- User ID-related error reports result from undefined user IDs or failed logon attempts resulting from incorrect user IDs, incorrect passwords, or incorrect new passwords.

The following table lists the commands for each security package that generates a user ID-related error report. Refer to your security package documentation for exact syntax and necessary authority.

Security Package	Report Command
RACF	RACFRW/SELECT VIOLATIONS/EVENT LOGON/LIST
CA–TOP SECRET	TSSUTIL DRC(PW) EVENT(VIOL)
CA-ACF2	ACFRPTPW (Invalid Password/Authority Log)

- Data set access error reports result from any data set access violations. All Connect:Enterprise resource checking is based on the use of pseudo data set names.

  The following table lists the commands for each security package that generates a data set access error report. Refer to your security package documentation for exact syntax and necessary authority.

Security Package	Report Command
RACF	RACFRW/SELECT VIOLATIONS/EVENT ACCESS/ CLASS(DATA SET)/LIST
CA–TOP SECRET	TSSUTIL DRC(DS) EVENT(VIOL)
CA-ACF2	ACFRPTDS (Data Set/Program Event Log)

Using Access Level Checking

To determine if a user has access to resources or if a resource is adequately protected, check access levels using the following security package commands. Refer to your security package documentation for exact syntax and necessary authority.

- Use the following security package commands to verify user access levels:

Security Package	Command
RACF	SEARCH and LISTDSD
CA–TOP SECRET	TSS LIST
CA-ACF2	ACFRPTRX (The Logon ID Access)

- Use the following security package commands to verify resource access levels:

Security Package	Command
RACF	LISTDSD

CA–TOP SECRET	TSS WHOHAS
CA-ACF2	ACFRPTXR (The Cross Reference)

✦ Use the following security package commands to test access levels:

Security Package	Command
RACF	LISTDSD
CA–TOP SECRET	TSSSIM
CA-ACF2	ACFTEST

Traces

Use the following commands to generate a trace and write reports based on trace data for each security package.

Note: Use of these security package functions require certain authority levels. Check with your security administrator if you do not have the proper authority to run a trace.

1. Use the following commands to generate trace data:

Security Package	Trace Command
RACF	ALTDSD
CA–TOP SECRET	TSS ADD TRACE
CA-ACF2	ACF CHANGE

2. Use the following commands to generate a report based on the trace data:

Security Package	Report Command
RACF	RACFRW/ USER/ LIST
CA–TOP SECRET	Trace is written directly to JOBLOG or JESLOG
CA-ACF2	ACFRPTST (the SAF Trace)

Understanding Security Interface Calls

This section describes the security calls for each security check. All security calls are the same regardless of the security package. The security calls execute only if all of the following occur:

- The security interface is activated by specifying one or more of the ODF security-related parameters, such as MBXSECURE or UIFSECURE
- Connect:Enterprise is executing in an authorized environment
- The previously called security exit (if it exists) returns a zero return code

Logon Security Calls

The logon security check uses the RACROUTE macro. Each logon password check requires two calls.

The first call establishes the security environment and validates the user ID and password. It has two forms, one with the new password field and one without.

The second call deletes the security environment and frees any allocated storage. The second call is done only if the first call results in a zero return code.

The following is a sample of the two security calls without the new password field:

```
RACROUTE REQUEST=VERIFY,ENVIR=CREATE,USERID=USERNAME,PASSWRD=PASSWORD,ACEE=WKACEE,
     WORKA=WKWORKA
RACROUTE REQUEST=VERIFY,ENVIR=DELETE,ACEE=WKACEE,WORKA=WKWORKA
```

The following is a sample of the two security calls with the new password field:

```
RACROUTE
REQUEST=VERIFY,ENVIR=CREATE,USERID=USERNAME,PASSWRD=PASSWORD,NEWPASS=NEWPSWRD,
     WORKA=WKWORKA,ACEE=WKACEE
RACROUTE REQUEST=VERIFY,ENVIR=DELETE,ACEE=WKACEE,WORKA=WKWORKA
```

Access is granted if the return code from the first call results in a return code of zero.

Batch/Function Security Calls

The batch/function security check uses the RACROUTE macro. Each batch/function check requires three calls. The first call establishes the security environment. The second call executes the security check. The third call deletes the security environment and frees any allocated storage. The third call executes only if the return code from the first call is zero.

The following is a sample of the three security calls for online Connect:Enterprise:

```
RACROUTE REQUEST=VERIFY,ENVIR=CREATE,USERID=USERNAME,ACEE=WKACEE,WORKA=WKWORKA,
             PASSCHK=NO
RACROUTE REQUEST=AUTH,ENTITY=(DSNAME),ATTR=UPDATE,CLASS=’DATA SET’,ACEE=(WKACEE),
             WORKA=WKWORKA
RACROUTE REQUEST=VERIFY,ENVIR=DELETE,ACEE=WKACEE,WORKA=WKWORKA
```

The following is a sample of the three security calls for offline utilities:

```
RACROUTE REQUEST=VERIFY,ENVIR=CREATE,USERID=USERNAME,ACEE=WKACEE,WORKA=WKWORKA,
      PASSCHK=NO
RACROUTE REQUEST=AUTH,ENTITY=(DSNAME),ATTR=UPDATE,CLASS='DATA SET',ACEE=(WKACEE),
      WORKA=WKWORKA
RACROUTE REQUEST=VERIFY,ENVIR=DELETE,ACEE=WKACEE,WORKA=WKWORKA
```

Access is granted if the return code from the first call results in a return code of zero.

VSAM Batch File Security Calls

The VSAM batch file security check uses the RACROUTE macro. The VSAM file check requires three calls. The first call establishes the security environment. The second call executes the security check. The third call deletes the security environment and frees any allocated storage. The third call executes only if the return code from the first call is zero.

The following is a sample of the three security calls:

```
RACROUTE REQUEST=VERIFY,ENVIR=CREATE,ACEE=WKACEE,WORKA=WKWORKA,USERID=USERNAME,
        PASSCHL=NO
RACROUTE REQUEST=AUTH,ENTITY=(DSNAME),ATTR=UPDATE,CLASS='DATA SET',ACEE=(WKACEE),
        WORKA=WKWORKA
RACROUTE REQUEST=VERIFY,ENVIR=DELETE,ACEE=WKACEE,WORKA=WKWORKA
```

Access is granted if the return code from the first call results in a return code of zero.

Using Logon and Batch Security for SNA and BSC Sessions Without the Connect:Enterprise Security Interface

You can use the SECURITY=LOGON|BATCH ODF *OPTIONS parameter for online security instead of, or in addition to, the Connect:Enterprise security interface. This following sections describe this parameter:

SECURITY=LOGON

SECURITY=LOGON verifies that all SNA remote sites include a remote name, predefined in the *REMOTE section of the ODF, when they log on to Connect:Enterprise. It further verifies that the remote logs on from a valid LU defined in either the *REMOTE entry or in a *POOL definition pointed to by the *REMOTE entry.

SECURITY=LOGON does not verify passwords; that is a security interface function. Also, it is not invoked for APPC logons (ISPF and CICS users). However, you can use SECURITY=LOGON in addition to the security interface to ensure that all SNA remote sites (including those using Gateway) are logged on to the Connect:Enterprise system from a known LU.

SECURITY=BATCH

SECURITY=BATCH requires that all transactions from remote sites supply a Mailbox ID that is predefined in the *SECURITY section of the ODF.

SECURITY=BATCH does not validate that a particular remote has access to a certain function against the Mailbox ID; that is a security interface function. Instead, it prevents any remote from accessing a Mailbox ID that is not listed in the *SECURITY section of the ODF. You can use it in addition to the security interface to prevent batches with unknown Mailbox IDs from being added to the data repository.

When you specify SECURITY=BATCH in the ODF, you must supply the valid Mailbox IDs in the ODF *SECURITY records. After the Mailbox IDs are defined in the ODF, remote sites with text editing capability can insert a record containing the ID in front of the data file.

Connect:Enterprise recognizes the valid Mailbox ID when it receives data from the host. If the host receives a batch without a Mailbox ID, Connect:Enterprise tries to determine it as follows:

- For SNA connections, Connect:Enterprise uses the remote name defined in the *REMOTES NAME parameter in the ODF.
- For BSC connections, Connect:Enterprise uses the remote name from the BSC SIGNON card, if specified, or the line ID as defined in the M$LINEX macro in the user assembly.

To force SNA remote sites to supply a valid Mailbox ID with each batch, omit their remote names from the *SECURITY section of the ODF.

To allow BSC remote sites on certain lines to access Connect:Enterprise without a BSC free-form SIGNON card, define the line ID (from M$LINEX) as one of the valid Mailbox IDs in the *SECURITY section of the ODF. However, specifying the line ID permits any remote sites accessing that line to send data without a Mailbox ID.

Example

The following example shows a network containing three remote sites: RMT001, RMT002, RMT003. RMT001 is used by a minicomputer to transmit only transparent binary data to the host and cannot insert a Mailbox ID in the front of its data files. RMT002 and RMT003 are remote personal computers with text editing capabilities. You can also assign Mailbox IDs to other remote personnel who have access to these PCs, and you can still use SECURITY=BATCH without restricting RMT001. The ODF *SECURITY record contains the following:

```
*SECURITY
  ID=TOM,ID=MARY,ID=BILL,ID=FRED,ID=ANN
  ID=JIM,ID=SAM,ID=JOAN,ID=JIMSMITH
  ID=MARIA
  ID=RMT001
```

In this example, RMT002 and RMT003 users must supply a valid Mailbox ID to submit or obtain data. RMT001 can send inbound batches without changes, since the Mailbox ID defaults to RMT001, which is valid.

For BSC connections, replace ID=RMT001 with the ID of the BSC line (from the M$LINEX assembly) that RMT001 uses, if the remote does not use the BSC free-form SIGNON. However, specifying the line ID in the *SECURITY section permits any remote using that line to send data without a Mailbox ID.

Chapter 11

Setting Up Connections to Other Communications Products

This chapter contains samples of Connect:Enterprise connections to other communications products, including:

- JES2 (BSC and SNA)
- LU Type 1 RJE devices
- IBM NPSI (X.25 network)
- VSE/Power (SNA and BSC)
- expEDIte/DIRECT

This chapter also includes instructions to set up Connect:Enterprise to communicate with the following Sterling Commerce products:

- Connect:Enterprise (SNA and BSC connections)
- Gentran Server Communication Module for OS/0400
- Connect:Tracs for MVS and VSE (BSC)
- Connect:Enterprise Gateway

Connect:Enterprise JES2 Support for BSC Sites

Connect:Enterprise sends and receives batches with JES2 as the remote site. Auto Connect lists for JES2 are defined in the same manner as other BSC remote sites, with a few additional operands for JES use only.

The following example shows two sample JES2 remote sites with the JES parameter settings in bold type:

```
*CONNECT
  LISTNAME=JESLIST
    TYPE=BSCAD
    TIME=02:00, 04:00
    JES=YES
    SIGNOFF=YES
    DISCINTV=300
      JES01 07 5551212 MODE=SENDRECV BLOCK=5 CMP=Y
/*SIGNON  RMT050 PASSWRD1
      JES02 07 5551313 MODE=SENDONLY BLOCK=5 CMP=Y
/*SIGNON RMT51 PASSWRD2
```

This Auto Connect list uses an auto dial line to dial the two remote sites daily at 2:00 and 4:00 a.m. Because JES=YES is specified, Connect:Enterprise uses appropriate I/O for JES2 communications. SIGNOFF=YES indicates that Connect:Enterprise should send the standard JES2 signoff before ending the JES connection.

A unique JES signon record is specified for each of the two remote sites immediately following the remote specification record. Connect:Enterprise first calls site JES01, sends all transmittable batches for the ID JES01 and for the listname JESLIST, then turns the line around to receive output from JES. Connect:Enterprise then calls the second JES remote, sending batches with the ID JES02 and for JESLIST. It does not receive output from JES02 because the mode is Send Only.

A disconnect interval of 300 seconds is specified to allow JES five minutes to process data and respond to Connect:Enterprise. Connect:Enterprise invokes the disconnect interval at three critical points in the JES communications:

- The disconnect interval is invoked when Connect:Enterprise sends an ENQ for permission to send to JES, just after the initial connection is made.
- The disconnect interval is invoked when Connect:Enterprise turns the line around to receive, immediately after all batches have been transmitted.
- The disconnect interval is invoked after each complete batch is received from JES, to allow multiple batches from JES.

When a disconnect interval is specified, I/O completion time-outs are ignored, and the appropriate I/O is reissued for the duration of the disconnect interval or until JES responds. However, the time-out console messages are still displayed by BTAM.

The JES2 parameters used when installing JES also include a DISCINTV parameter, which is similar to that used for Connect:Enterprise. If Connect:Enterprise does not respond to JES for the specified interval, JES terminates the connection with Connect:Enterprise. If the Connect:Enterprise disconnect interval is not working as expected, check the setting for the DISCINTV subparameter used for the remote site in the JES parameters to ensure that JES is not prematurely ending the connection.

Note: The DISCINTV parameter in Connect:Enterprise is specified in seconds, while the DISCINTV parameter in JES is specified in units of 32 seconds. For example, to set the DISCINTV parameter to 64 seconds for both Connect:Enterprise and JES, you would set the parameter to 64 in Connect:Enterprise and to 2 in JES.

Because Connect:Enterprise relies on BTAM time-outs when accumulating the elapsed time in a disconnect interval, the actual disconnect interval has a tolerance level. If a BTAM time-out occurs after 28 seconds and the Connect:Enterprise disconnect interval is set at 30 seconds, then the connection does not end until the second BTAM time-out occurs at 56 seconds.

JES Requirements

JES2 must be properly installed on the system before Connect:Enterprise can initiate JES transmissions. Refer to the appropriate JES2 installation manual.

Note the remote names and line passwords defined in the JES2 parameters for proper specification of the JES signon to Connect:Enterprise.

Use caution when defining the DISCINTV in the JES2 parameters. The value is specified differently from the DISCINTV defined to Connect:Enterprise. If Connect:Enterprise is to control the disconnect interval during JES communications, ensure that the JES2 parameters do not conflict with the Connect:Enterprise parameters.

JES Transparency

To allow Connect:Enterprise to receive transparent punch data from JES, a line must be defined as transparent (TRANSP) in the JES parameters.

Specify the proper telephone number for the transparent line in the Auto Connect list. Connect:Enterprise cannot properly extract a transparent batch from JES unless the data begins with the correct transparency control characters (DLE, STX).

It is not necessary to use TRANSPAR=YES in the Connect:Enterprise Auto Connect list to receive transparent punch data from JES. Connect:Enterprise can receive transparent punch data from JES that begins with the Connect:Enterprise $$ADD record. This record is removed from the batch before the data is stored on the VSAM Batch Files, and it must always be exactly 80 characters in length.

To send transparent data from Connect:Enterprise to JES, specify TRANSPAR=YES in the Auto Connect list.

Connect:Enterprise JES2 Support for SNA Sites

Connect:Enterprise can have a session with a JES2 site with the following restrictions:

- Connect:Enterprise must initiate the session with an Auto Connect session.
- Connect:Enterprise must act as the Secondary Logical Unit (SLU) in the session.

- Connect:Enterprise must pass USERDATA containing the JES2 remote name and the password if required.
- Each batch returned to Connect:Enterprise from JES2 is added as AC BATCH WITHOUT $$ADD under the Mailbox ID of the Connect:Enterprise remote name, unless a $$ADD control record has been received.
- The JES2 remote definition (in JES2PARMS) must specify LUTYPE1, NOCMPCT, COMP, and cannot specify SETUPHDR (for pre-JES2 Version 3.1.1 releases) or SETUP=PDIR (for post-JES2 Version 3.1.1 releases).
- SUSPEND/RESUME FMHs are not supported.
- There is no MLU (Multiple Logical Unit) support.

Sample Implementation

In this example, Connect:Enterprise (ENTPRSA) acts as the SLU and JES2 acts as the Primary Logical Unit (PLU). First, a file is added to the repository as Mailbox ID JES2JOB with the following information:

```
//MYJOB001 JOB (acctg info), ’NAME’,CLASS=X,NOTIFY=userid
/*ROUTE PUNCH RMT15
//COPY      EXEC PGM=IEBGENER
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT2    DD SYSOUT=B /* PUNCH OUTPUT */
//SYSUT1    DD *
DATA RECORD ONE    FIRST RECORD
DATA RECORD TWO    MIDDLE RECORD
DATA RECORD THREE  LAST RECORD
```

The operator starts the session by entering the $$CONNECT console command. ENTPRSA sends the batch JES2JOB to JES2 remote RMT15. The DISCINTV on the Auto Connect list ends the session.

The following illustration shows the Connect:Enterprise ODF setup and the related JES2 definitions for this example:

```
  ENTPRSA (SLU)                               JES2 (PLU)
                                                                              72
*OPTIONS                             RMT15   LUTYPE1,LUNAME=XXXXXXXX,          +
  VTAM=YES                                   NUMPR=1,NUMPU=1,NUMRD=1,          +
  VPF='CE.VPF'                               NOCON,SETUPMSG,NOCMPT,            +
  MODIFY=YES                                 SETUPINF,PASSWORD=PSWD,           +
  CONSLOG=YES                                COMP
  PASSWORD=MYPASS                            R15.RD1 PRIOLIM=9
  APPLID=CE A                                R15.PR1 NOSEP,PRWIDTH=255,        +
  VSESSLIM=8                                 WS=(W,R,Q,PMD,LIM,F/P),           +
  CMB001I='ENTPRSA AS SLU'                       SELECT=PRINT,EJECT=NO
**                                           R15.PU1 NOSEP,SELECT=PUNCH,       +
*CONNECT                                     LRECL=255
**
  LISTNAME=SEND2JES
    TYPE=LU1RJE
    DISCINTV=30
    ACSESS#=1
    JES2JOB MEDIA=PU
**
*REMOTES
**
  NAME=JES2JOB
    TYPE=LU1RJE
    RMTACB=JES2
    LOGMODE=RJE3770X
    CONSOLE=NO
** JES2 PASSWORD IS OPTIONAL **
    USERDATA='RMT15,,PASSWORD'
    MEDIA=PU
```

Note: Do not use MEDIA=PR for the remote. The APPL definition for Connect:Enterprise must specify PARSESS=YES.

Connect:Enterprise Connections with SNA LU Type 1 RJE Devices

This section provides a sample NCP generation to help you connect an SNA LU Type 1 RJE device with Connect:Enterprise. Usually these devices have already been included in your NCP generation for use with JES2, and the requirements for Connect:Enterprise are identical to those for JES2. The installation instructions that accompany your remote devices explain how to define them in your NCP generation. The NCP generation includes line specifications for your dedicated lines and for your switched lines.

If you are using the Connect:Enterprise Auto Connect function for host-initiated calls to remote sites on switched lines, you must specify CALL=INOUT on one or more LINE statements for switched lines. *This is the main Connect:Enterprise requirement in the NCP generation.* If your network is already generated to support LU Type 1 RJE devices with JES2, other changes are probably not necessary.

The following is a sample NCP generation showing a switched line:

```
       TITLE ’IBM 3705 PEP GEN’
       SPACE 3
****************************************************************
*     3705 DEF FOR DIAL-IN AND DIAL-OUT SDLC LINE              *
*     Connect:Enterprise SNA ENVIRONMENT                   *
****************************************************************    72
SDLCSW01   GROUP   DIAL=YES,        SWITCHED LINES                   x
                   LNCTL=SDLC,      SDLC LINE TYPE                   x
                   POLLED=YES,      SDLC LINE                        X
                   REPLYTO=1.0,                                      x
                   TYPE=NCP
L022SW     LINE    ADDRESS=022,     LINE ADDRESS IN IBM 3705         x
                   SPEED=4800,      LINE SPEED                       x
                   CLOCKING=EXT     EXTERNAL MODEM CLOCKING          x
                   DUPLEX=HALF,     HALF DUPLEX COMMUNICATION LINE   x
                   CALL=INOUT       DIAL-IN/OUT LINE                 x
                   AUTO=(020),      AUTO CALL UNIT                   x
                   INTPRI=2,                                         x
                   ISTATUS=ACTIVE,                                   x
                   NRZI=NO,                                          x
                   RETRIES=(7,4,5),  NUMBER OF ATTEMPTS TO RECOVER   x
                   TRANSFR=16        MAX NUMBER OF BUF FOR MESSAGES
P022SW     PU      MAXLU=16
L023SW     LINE    ADDRESS=023,      LINE ADDRESS IN IBM 3705        x
                   SPEED=4800,       LINE SPEED                      x
                   CLOCKING=EXT,     EXTERNAL MODEM CLOCKING         x
                   DUPLEX=HALF,      HALF DUPLEX COMMUNICATION LINE  x
                   CALL=INOUT,       DIAL-IN/OUT LINE                x
                   INTPRI=2,                                         x
                   ISTATUS=ACTIVE,                                   x
                   NRZI=NO,                                          x
                   RETRIES=(7,4,5), NUMBER OF ATTEMPTS TO RECOVER    x
                   TRANSFR=16        MAX NUMBER OF BUF FOR MESSAGES
P023SW  PU         MAXLU=16
        SPACE 3
```

Your installation may require different generation macros and line addresses. Consult your *IBM 37XX Control Program Generation and Utilities Guide and Reference Manual* for additional information.

The following is a sample NCP generation for a leased line:

```
*****************************************************************
*        3725  DEF FOR SDLC lease (nonswitched) LINE            *
*            SNA ENVIRONMENT/RJE 3770                           *
*****************************************************************   72
SDLCNS01   GROUP   LNCTL=SDLC,       SDLC LINE TYPE                  x
                   TYPE=NCP,                                         x
                   CLOCKING=EXT,     EXTERNAL MODEM CLOCKING         x
                   SPEED=9600,       LINE SPEED                      x
                   DUPLEX=HALF,      HALF DUPLEX COMMUNICATION LINE  x
                   NRZI=NO,                                          x
                   REPLYTO=1.0,                                      x
                   RETRIES=(7,4,5)
L016NS     LINE    ADDRESS=(016,HALF), LINE ADDRESS IN IBM 3725      x
                   SPEED=9600,       LINE SPEED                      x
                   SERVLIM=4,                                        x
                   MAXPU=1,                                          x
                   VPACING=(14,1)
SER016     SERVICE ORDER=(P016NS),                                   x
                   MAXLIST=1
P016NS     PU      ADDR=01,                                          x
                   PUTYPE=2,                                         x
                   DISCNT=(NO),                                      x
                   MAXDATA=265,                                      x
                   MAXLU=04,                                         x
                   MAXOUT=7,                                         x
                   MODETAB=RJE3770,                                  x
                   PACING=(7,1),                                     x
                   PASSLIM=20,                                       x
                   SSCPFM=USSSCS,                                    x
                   USSTAB=UTABSNA,                                   x
                   VPACING=14
LU016NS0   LU  LOCADDR=2
LUO16NS1   LU  LOCADDR=3
LUO16NS2   LU  LOCADDR=4
```

Connect:Enterprise Connections with NPSI

Connect:Enterprise can support sessions with an X.25 network using the IBM Network Control Program Packet Switching Interface (NPSI). To accomplish this, Connect:Enterprise uses the NPSI supported 3767 protocol.

This requires NPSI V3.R3 or later. This enables inbound batch separation using standard SNA chaining. This also enables a separate translate table for the X.25 line (if required).

Implementation Considerations

When planning an X.25 connection to Connect:Enterprise using NPSI, consider the following questions:

✦ What will be on the other side of the X.25 connection (PC, host)?

If the remote device is not an IBM mainframe, translation issues have to be addressed.

- What data format do the remote sites use (EBCDIC, ASCII)?

  If the remote is an ASCII-based computer, an NPSI translation table is required.
- What record and file separation does the remote site use?

  Outbound, Connect:Enterprise separates records with a single character (usually X’1E’). DOS-based PCs use two characters CR/LF. The NPSI translate table in the example translates the IRS (X’1E’) to a CR. Most PC ASYNC transmission packages can translate the inbound CR to a CR/LF. The process is reversed for data inbound to NPSI and Connect:Enterprise.
- Does the remote site support LU1 3770 compression?

  You probably have to turn compression off for your NPSI remote sites.
- What routing and session information will the X.25 network require from each site to ensure end-to-end connectivity?

  There are other considerations. Some are addressed by the X.25 network. When you need information, consider the type of information you require and call the technical support group for that product.

X.25 Connectivity Example

In the example that follows, NPSI connects Connect:Enterprise to an X.25 network. The remote sites are IBM compatible PCs running an asynchronous transmission program capable of CR-to-CR/LF conversion. The example includes the following samples:

- NPSI/NCP code
- VTAM Switched Network (SWNET) definition
- VTAM USSTAB
- Portion of a Connect:Enterprise ODF

NPSI/NCP Code

The following is an example of NPSI and NCP code. Refer to the IBM NPSI manuals for more information

```
                                                                     72
PEPGEN   BUILD  BFRS=128,            OPTIMUM LINE USAGE               X
                 X25.MCHCNT=3,        X25 # OF PHY CIRCUITS           X
                 X25.MAXPIU=4096,     X25 MAX INBOUND PIU LENGTH      X
                 X25.PAHINDX=1,       X25 INDEX POINTER TO PAD PARM   X
                 X25.SNAP=YES,        X25 TURNS ON SNAP SHOT          X
                 X25.USGTIER=4        X25 NPSI USAGE TIER
 *
          LUDRPOOL   NUMTYPE1=3, NUMTYPE2=3     #OF NPSI  LU’S
*****************************************************************
*               X 2 5 P A D       M A C R O                     *
*****************************************************************
PAD1     X25.PAD INDEX=1,                   ENTRY OF PAD TABLE
                 PADPARM=0715080001000200   PAD PARMS SENT TO NTWRK
*
****************************************************************
*              X 2 5 T R A N      M A C R O                    *
*  (A-E / E-A TABLE) 0 1 2 3 4 5 6 7 8 9 A B C D E F           *
****************************************************************
TRANS1   X25.TRAN USER=1,                                             X
                DCIN0=00010203372D2E2F1605250B0C1E0EOF,               X
                DCIN1=101112133C3D322618193F271C1D1E1F,               X
                DCIN2=405A7F7B5B6C507D4D5D5C4E6B604B61,               X
                DCIN3=F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F,               X
                DCIN4=7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6,               X
                DCIN5=D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D,               X
                DCIN6=79818283848586878889919293949596,               X
                DCIN7=979899A2A3A4A5A6A7A8A9C06AD0A1FF,               X
                DCIN8=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCIN9=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCINA=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCINB=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCINC=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCIND=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCINE=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCINF=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,               X
                DCOT0=404040404040404040404000C0D4040,               X
                DCOT1=4040404000D40404040404040400D40,               X
                DCOT2=4040404000A40404040404040404007,               X
                DCOT3=40404040404040044040404040404040,               X
                DCOT4=2040404040404040404040402E3C282B7C,               X
                DCOT5=26404040404040404040421242A293B5E,               X
                DCOT6=2D2F4040404040404407C2C255F3E3F,               X
                DCOT7=40404040404040404403A2340273D22,               X
                DCOT8=40616263646566676869404040404040,               X
                DCOT9=406A6B6C6D6E6F707172404040404040,               X
                DCOTA=407E737475767778797A404040404040,               X
                DCOTB=40404040404040404040404040404040,               X
                DCOTC=7B414243444546474849404040404040,               X
                DCOTD=7D4A4B4C4D4E4F505152404040404040,               X
                DCOTE=5C405354555657585955A404040404040,               X
                DCOTF=30313233343536373839404040404040
*                      0 1 2 3 4 5 6 7 8 9 A B C D E F
```

Continued

```
******************************************************************
*                                                                   72
*               X 2 5 N E T        M A C R O                     *
******************************************************************
           X25.NET DM=N0,    DISCONNECT MODE IN LAPB                 X
                 CPHINDX=1,   GREATEST INDEX VALUE                   X
                 OUHINDX=1,   GREATEST INDEX VALUE                   X
                 NETTYPE=1    TYPE 1 NETWORK
******************************************************************
*               X 2 5 V C C P T    M A C R O                     *
******************************************************************
           X25.VCCPT INDEX=1,   INDEX IN V C CONN PARM TABLE         X
                 MAXPKTL=128,   MAX PACKET LENGTH                    X
                 INSLOW=(25,0), FREE BUFFER UNSAFE/DANGER SITUATION  X
                 VWINDOW=2
******************************************************************
*               X 2 5 O U F T      M A C R O                     *
******************************************************************
           X25.OUFT INDEX=1     INDEX IN OPT USER FACILITIES TABLE
******************************************************************
*               X 2 5 M C H        M A C R O                     *
******************************************************************
           X25.MCH ADDRESS=XXX,       LINE INTERFACE ADDRESSES       X
                 FRMLGTH=131,         MAX FRAME LENGTH IN BYTES      X
                 LCGDEF=(0,20),      (LOG-CH-GRP,LOG-CH-HI-NUM)      X
                 MWINDOW=7,           FRAME WINDOW SIZE              X
                 ANS=CONTINUE,                                       X
                 DBIT=NO,             DELIVERY CONFIRMATION BIT      X
                 GATE=NO,             GATE OR DATE NOT USED          X
                 LCNO=NOTUSED,        LOG CHANNEL 0 NOT USED         X
                 LLCLIST=LLC5,        SVC TYPE                       X
                 PUNAME=XPUXXX,       PUNAME FOR THIS LINK           X
                 LUNAME=XLUXXX,       SPECIFY LUNAME                 X
                 MBITCHN=YES,         RU CHAINING IS SUPPORTED       X
                 NCPGRP=X25LSXXX,     NCP GROUP MACRO NAME           X
                 NDRETRY=3,       TIMES NP/TP SEQUENCE RETRY COUNT   X
                 NPRETRY=31,      TP TIMEOUT RECOVERY RETRAN TIMES   X
                 PAD=INTEG,           INTEGRATED PAD                 X
                 TRAN=USER1,      USER SET TRANS TBL FOR EVEN PARITY X
                 PKTMODL=8,           MODULO 8 FOR PACKETS           X
                 STATION=DTE,         THIS IS DTE END                X
                 TDTIMER=3,       DELAY BETWEEN ND RETRANS-SECONDS   X
                 ISTATUS=ACTIVE,                                     X
                 TPTIMER=6.0          TI TIMER IN SECONDS
******************************************************************
*               X 2 5 L C G        M A C R O                     *
******************************************************************
           X25. LCG LCGN=0     LOGICAL CHANNEL GROUP # FOR SUBSEQ VCS
******************************************************************
*               X 2 5 V C        M A C R O                       *
******************************************************************
           X25.VC LCN=(1,3),   LOG CHAN #0 NOT USED                  X
                 TYPE=S,       SWITCHED  CIRCUIT                     X
                 OUFINDX=1,  INDEX IN FACILITIES TABLE FROM X25OUFT  X
                 VCCINDX=1,  INDEX IN PARAMETER TABLE FROM X25VCCPT  X
                 CALL=IN,      REMOTE DTE INITIATES CALLS            X
                 COMMITO=4,    IDLE VC COMMIT BUFFER TIMEOUT         X
                 NCPGRP=X25SWXXX,  GROUP MACRO NAME                  X
                 RETVCCT=3,  RETRANS OF PHYSICAL SERV CMDS IF SNA    X
                 ISTATUS=ACTIVE,                                     X
                 RETVCTO=30  TIME BETWEEN RETRANS OF PHYS SERV CMDS
*
          X25.END
```

VTAM SWNET

The Connect:Enterprise VTAM system requires a switched network definition. The IDNUM parameter is determined from the NPSI GEN output as shown in the following example:

```
                                                                    72
SWNAME   VBUILD TYPE=SWNET
PUNPSI1 PU     IDNUM=00006,MODETAB=NPSIMODE,USSTAB=USNP01,          X
               ADDR=01,IDBLK=003,MAXOUT=1,IRETRY=YES,PUTYPE=1,      X
               DISCNT=(YES,F),PASSLIM=1,                            X
               MAXDATA=265,PACING=1,ISTATUS=ACTIVE
LUNPSI1 LU     LOCADDR=0,BATCH=NO,PACING=1,VPACING=2,TERM=TWX,      X
               SSCPFM=USSNTO
PUNPSI2 PU     IDNUM=00004,MODETAB=NPSIMODE,USSTAB=USNP01,          X
               ADDR=01,IDBLK=003,MAXOUT=1,IRETRY=YES,PUTYPE=1,      X
               DISCNT=(YES,F),PASSLIM=1,                            X
               MAXDATA=265,PACING=1,ISTATUS=ACTIVE
LUNPSI2 LU     LOCADDR=0,BATCH=NO,PACING=1,VPACING=2,TERM=TWX,      X
               SSCPFM=USSNTO
PUNPSI3 PU     IDNUM=00002,MODETAB=NPSIMODE,USSTAB=USNP01,          X
               ADDR=01,IDBLK=003,MAXOUT=1,IRETRY=YES,PUTYPE=1,      X
               DISCNT=(YES,F),PASSLIM=1,                            X
               MAXDATA=265,PACING=1,ISTATUS=ACTIVE
LUNPSI3 LU     LOCADDR=0,BATCH=NO,PACING=1,VPACING=2,TERM=TWX,      X
               SSCPFM=USSNTO
```

VTAM USSTAB

Since X.25 remote sites cannot provide Connect:Enterprise with their remote names at logon, a custom USSTAB table is required for each remote, as shown in the following example:

```
USNP1    USSTAB FORMAT=DYNAMIC
* USS TABLE FOR X.25 NPSI ACCESS TO ENTPRS
*
  USSCMD  CMD=LOGON,REP=LOGON,FORMAT=PL1
  USSPARM PARM=P1,REP=APPLID,DEFAULT=ENTPRS
  USSPARM PARM=LOGMODE,DEFAULT=XXXXXXXX
  USSPARM PARM=DATA,DEFAULT='NPSI01'
*
* LOGOFF COMMAND
*
  USSCMD  CMD=LOGOFF,FORMAT=PL1
  USSPARM PARM=APPLID
  USSPARM PARM=TYPE,DEFAULT=UNCOND
  USSPARM PARM=HOLD,DEFAULT=YES
*
END      USSEND
         END
```

ODF Configuration for NPSI

The following example shows the ODF *REMOTES section for an NPSI connection:

```
*REMOTES
  NAME=NPSI01
    TYPE=LU1RJE
    LUNAME=LUNPSI1
    MEDIA=PR
    BLKSIZE=256
    COMPRESS=NO
    FMH=X25
  NAME=NPSI02
    TYPE=LU1RJE
    LUNAME=LUNPSI2
    MEDIA=PR
    BLKSIZE=256
    COMPRESS=NO
    FMH=X25
  NAME=NPSI03
    TYPE=LU1RJE
    LUNAME=LUNPSI3
    MEDIA=PR
    BLKSIZE=256
    COMPRESS=NO
    FMH=X25
```

Connect:Enterprise POWER Support for SNA Sites

Connect:Enterprise has the following session restrictions with POWER:

- Connect:Enterprise must initiate the Auto Connect session.
- Connect:Enterprise must act as the SLU in the session.
- Connect:Enterprise must pass USERDATA containing the POWER remote name and the password if required.
- Each batch returned to Connect:Enterprise from POWER is added as AC BATCH WITHOUT $$ADD under the Mailbox ID of the Connect:Enterprise remote name, unless a $$ADD control record has been received.
- SUSPEND/RESUME FMHs are not supported.
- FMH Type2 PDIR records are ignored by Connect:Enterprise.
- Multiple Logical Units (MLUs) are not supported.

Sample Implementation

In this example, Connect:Enterprise (ENTPRSA) acts as the SLU and POWER acts as the PLU. First, a file is added to the repository as Mailbox ID JES2JOB with the following information:

```
* .. START LST,A
* .. START PUN,A
* $$ JOB JNM=PWERJOB,DISP=D,CLASS=A
* $$ LST CLASS=A
// JOB PWERJOB LIST ENTPRS LIBRARY
*
// EXEC LIBR
   LISTDIR S=ENTPRS.BASE
/*
/&
* $$ EOJ
/*
```

The operator starts the session by entering a $$CONNECT console command. ENTPRSA sends the data batch (JOB, containing the following JOB JCL) to POWER. This data batch is stored on the ENTPRSA VSAM files under the following Mailbox ID PWRJOB.

POWER receives and executes the JOB. The JOB output is returned to Connect:Enterprise as batches through RMT15. Connect:Enterprise creates a new batch each time it receives BDS FMH. The DISCINTV on the Auto Connect list ends the session.

The Connect:Enterprise ODF and the parameters required in POWER are following:

```
   ENTPRSA (SLU)                               POWER (PLU)
                                                                                72
*OPTIONS                                  RMT 15 PRMT REMOTE=15                  +
  VPF=’ENTPRSA.VPF’                               TYPE=LUT1,                     +
  VTAM=YES                                        CONSOLE=YES
  MODIFY=YES
  CONSLOG=YES
  PASSWORD=MYPASS
  APPLID=ENTPRSA
  VSESSLIM=8
  CMB001I=’ENTPRSA AS SLU’
**
*CONNECT
**
  LISTNAME=SEND2PWR
    TYPE=LU1RJE
    DISCINTV=30
    ACSESS#=1
      PWRJOB MEDIA=PU
**
*REMOTES
**
  NAME=PWRJOB
    TYPE=LU1RJE
    RMTACB=POWER
    LOGMODE=RJE3770X
    CONSOLE=NO
** POWER PASSWORD IS OPTIONAL **
    USERDATA=’15,,PASSWORD’
      MEDIA=PU
```

Note: Do not use MEDIA=PR for the remote. The APPL definition for Connect:Enterprise must specify PARSESS=YES.

Connect:Enterprise POWER Support for BSC Sites

Connect:Enterprise sends and receives batches with POWER as the remote site. Auto Connect lists for POWER are defined in the same manner as other BSC remote sites, with a few additional operands for POWER use only.

The following figure shows two sample POWER remote sites with the POWER parameter settings in bold:

```
*CONNECT
  LISTNAME=PWRLIST
    TYPE=BSCAD
    TIME=02:00, 04:00
    POWER=YES
    SIGNOFF=YES
    DISCINTV=300
      POWER01 07 5551212 MODE=SENDRECV
* .. SIGNON3
      POWER02 07 5551313 MODE=SENDONLY
* .. SIGNON4
```

This Auto Connect list uses an auto dial line to dial the two remote sites daily at 2:00 and 4:00 a.m. Because POWER=YES is specified, Connect:Enterprise uses appropriate I/O for POWER communications. SIGNOFF=YES indicates that Connect:Enterprise sends the standard POWER signoff before ending the connection with POWER.

A unique POWER signon record is specified for each of the two remote sites immediately following the remote specification record. Connect:Enterprise first calls site POWER01, sends all transmittable batches for the ID POWER01 and the listname PWRLIST, then turns the line around to receive output from POWER. Connect:Enterprise then calls the second POWER remote, sending batches with the ID POWER02 and for PWRLIST. A receive is not done for POWER02 since the mode is Send Only.

A disconnect interval of 300 seconds is specified to allow POWER 5 minutes to process and respond to Connect:Enterprise. Connect:Enterprise invokes the disconnect interval at three critical points in the POWER communications:

- The disconnect interval is activated when Connect:Enterprise is sending ENQ to ask permission to send to POWER, immediately after the initial connection is made.
- The disconnect interval is activated when Connect:Enterprise turns the line around to attempt to receive, immediately after all batches have been transmitted.
- The disconnect interval is activated after each complete batch is received from POWER, to allow the receipt of multiple batches from POWER.

When a disconnect interval is specified, I/O completion time-outs are ignored and the appropriate I/O is reissued for the duration of the disconnect interval or until POWER responds. However, the time-out console messages are still displayed by BTAM.

POWER Requirements

POWER must be properly installed on the system before Connect:Enterprise can initiate POWER transmissions. Refer to the appropriate POWER installation manual.

Make a note of the remote names and line passwords defined in the POWER parameters for proper specification of the POWER signon to Connect:Enterprise.

Connect:Enterprise Connections with expEDIte/DIRECT (SNA Only)

To make a connection with the IBM expEDIte/DIRECT, follow these guidelines:

- The APPLID for expEDIte/DIRECT is IBM0DI01. This APPLID is a parameter in the ODF to set up a session. Register an APPL to APPL session between your Connect:Enterprise and expEDIte/DIRECT. Your APPLID is set up by your VTAM systems programmer and is a parameter in the ODF, *OPTION, APPLID=xxxxxxxx.
- Ensure that you have an IINUSERID, IEUSERID, and IEPSWD. If you do not have these, please request them from your ADVANTIS representative.
- Build an ODF using the SLU/PLU capability.
- Once the APPL to APPL registration process is complete, test the connection between Connect:Enterprise and expEDIte/DIRECT.

If you need help with expEDIte/DIRECT, obtain the IBM expEDIte/DIRECT documentation. Also, you can obtain expEDIte/DIRECT information from your ADVANTIS representative. Ensure that the registration process used by the ADVANTIS representative is an application to application LU1 type connection, not a SNUF type connection.

expEDIte/DIRECT Example

The following is a sample *CONNECT and *REMOTE entry:

```
*REMOTE
  NAME=IINIE
    TYPE=LU1RJE
    LOGMODE=RJE3770X
    BLKSIZE=256
    CONSOLE=NO
    COMPRESS=NO
    MEDIA=PU
    FMH=IE
    RMTACB=IBM0DI01
    USERDATA='ieuserid'
*CONNECT
  LISTNAME=SEND2IE
    TYPE=LU1RJE
    (any other parameters)
      IINIE IDLIST=IELOGON,IBMDATA,IEEDILOG
```

Note the following:

- The RJE3770X logmode entry is required.
- The FMH type connects to the IBM application.
- The value for the expEDIte/Direct application is IBM0DI01.
- USERDATA represents the remote to which your logon is sent and is the information exchange user ID.

To understand this example, the commands to logon to the IBM application are in the batch IELOGON. The data sent to the IBM application is in the batch IBMDATA. The commands to receive anything from the IBM application, including any log information, are in the batch IEEDILOG.

The IELOGON batch starts with the following:

```
IELOGON ACCOUNT(iinaccount) USERID(ieuserid) PASSWORD(iepswd) CHARGE(3);
```

The IEEDILOG batch can end with the following:

```
RECEIVE CLASS(EDILOG);
/*LOGOFF
```

IDLIST is not required. The logon commands, the data, and the additional IE commands can be concatenated when the STOUTL utility is run. A single batch would be created with the Mailbox ID of the remote entry. In the example, the remote ID is IINIE, so the STOUTL add uses the parameter ID=IINIE. For more information, refer to the STOUTL utility.

Connect:Enterprise-to-Connect:Enterprise Sessions (SNA)

Connect:Enterprise can support an infinite number of Connect:Enterprise-to-Connect:Enterprise sessions. These sessions require that Connect:Enterprise be used on both sides of the session with the proper VTAM definitions. In a Connect:Enterprise-to-Connect:Enterprise session, one side is the primary (PLU) and the other side is the remote (SLU). Either side (PLU or SLU) can start the session, but sessions can be started only by using the Auto Connect function. Further, a Connect:Enterprise can act as the PLU for some sessions and as the SLU for other sessions at the same time.

When Connect:Enterprise is acting as the remote site (SLU), the VSESSLIM parameter determines the maximum number of sessions. When Connect:Enterprise acts as the host site (PLU), as many sessions as desired can be started. CONSOLE=NO must be specified for all remote sites used in Connect:Enterprise-to-Connect:Enterprise sessions.

ODF Parameters

The following parameters are used in the ODF *REMOTES section for Connect:Enterprise-to-Connect:Enterprise communications:

Parameter	Description
RMTACB	This parameter is required if Connect:Enterprise is to act as the SLU in a session. It provides the name of the remote ACB for which a REQSESS is issued. The Connect:Enterprise with this parameter is the SLU when the Auto Connect session is started. The name coded must match the APPLID of the Connect:Enterprise that is the session partner and acts as the PLU. RMTACB is mutually exclusive with LUNAME. The absence of RMTACB indicates that this Connect:Enterprise is to act as the PLU for an Auto Connect session.
USERDATA	This parameter is required if this remote is used in an Auto Connect session and it is to go into session with another Connect:Enterprise. USERDATA specifies the remote name to be used in the Connect:Enterprise that is the session partner. This parameter must be enclosed in single quotation marks. The maximum length is 27.
LOGMODE	This optional parameter specifies the LOGMODE to be used for the session; it only has meaning if Connect:Enterprise is operating as the SLU.

Communications Requirements

The following items are required:

- Connect:Enterprise must specify PARSESS=YES on the APPL statement in the VTAM definition. (PARSESS enables multiple parallel sessions between two or more APPLS.)
- Connect:Enterprise-to-Connect:Enterprise operates as an APPL-to-APPL session; therefore, the proper network definitions and path tables must be in place.
- CONSOLE=NO is required in the remote definition of both Connect:Enterprise systems.
- If DISCINTV is used, specify it only on one side of the session.
- There is no MLU support for Connect:Enterprise-to-Connect:Enterprise; however, parallel sessions are allowed.

Sample Implementations

The following examples are shown:

- In Example 1, an SNA connection with two Connect:Enterprise systems, ENTPRSA acts as the SLU and sends three batches to ENTPRSB, which acts as the PLU.
- In Example 2, an SNA connection with two Connect:Enterprise systems, ENTPRSA (SLU) requests a batch from ENTPRSB (PLU).
- In Example 3, a BSC connection with two Connect:Enterprise systems, ENTPRSA sends a batch and requests a batch from ENTPRSB.

Example 1: PLU Sends Batches

ENTPRSA starts the sessions, acts as the SLU, and sends batches from three different remote sites on three different sessions at the same time to ENTPRSB. The sessions are started by the operator entering a $$CONNECT command and ended by the DISCINTV on the Auto Connect list (ENTPRSA). The ENTPRSA and ENTPRSB ODFs are shown:

```
       ENTPRSA (SLU)                    ENTPRSB (PLU)
*OPTIONS                             *OPTIONS
  VTAM=YES                             VTAM=YES
  VPF=’ENTPRSA.VPF’                     VPF=’CEENTPRSB.VPF’
  CONSLOG=YES                          CONSLOG=YES
  APPLID=ENTPRSA                          APPLID=ENTPRSB
  VSESSLIM=8                           VSESSLIM=8
  CMB001I=’ENTPRSA AS SLU’                CMB001I=’ENTPRSB AS PLU’
**                                     **
*CONNECT                              *REMOTES
**                                    **
  LISTNAME=SND3BCHS                     NAME=RMT4PLU1
    TYPE=LU1RJE                           TYPE=LU1RJE
    ACSESS#=3                             CONSOLE=NO
    DISCINTV=20                           MEDIA=PU
      JUNKP1 MEDIA=PU                 **
      JUNKP2 MEDIA=PU                   NAME=RMT4PLU2
      JUNKP3 MEDIA=PU                     TYPE=LU1RJE
**                                        CONSOLE=NO
*REMOTES                                  MEDIA=PU
**                                    **
  NAME=JUNKP1                           NAME=RMT4PLU3
    TYPE=LU1RJE                           TYPE=LU1RJE
    CONSOLE=NO                            CONSOLE=NO
    RMTACB=ENTPRSB                           MEDIA=PU
    USERDATA=’RMT4PLU1’
**
  NAME=JUNKP2
    TYPE=LU1RJE
    RMTACB=ENTPRSB
    CONSOLE=NO
    USERDATA=’RMT4PLU2’
**
  NAME=JUNKP3
    TYPE=LU1RJE
    RMTACB=ENTPRSB
    CONSOLE=NO
    USERDATA=’RMT4PLU3’
```

The batches of data sent from ENTPRSA (IDs JUNKP1, JUNKP2, and JUNKP3) must have a $$ADD card as the first record with the appropriate parameters as shown in the following example:

```
$$ADD ID=xxxxxxxx BID=’yyyyyyyyyyyyy’ MX=YES
DATA RECORD ONE
DATA RECORD TWO
...............
END OF DATA
```

If no $$ADD card is present in the data, the batch is received by ENTPRSB as a BATCH WITHOUT $$ADD for RMT4PLU1, 2, or 3, depending on whether it was sent from JUNKP1, 2, or 3.

Example 2: SLU Requests Batches

ENTPRSA starts the session, acts as the SLU, and requests a batch from ENTPRSB. The session is started at 10:00 a.m. by the Auto Connect timer function and stopped when the Auto Connect session ends.

The ODFs for ENTPRSA and ENTPRSB are shown:

```
       ENTPRSA (SLU)                    ENTPRSB (PLU)
*OPTIONS                             *OPTIONS
  VTAM=YES                              VTAM=YES
  VPF=’ENTPRSA.VPF’                      VPF=’ENTPRSB.VPF’
  CONSLOG=YES                           CONSLOG=YES
  APPLID=ENTPRSA                           APPLID=ENTPRSB
  VSESSLIM=8                            VSESSLIM=8
  CMB001I=’ENTPRSA AS SLU’                 CMB001I=’ENTPRSB AS PLU’
**                                    **
*CONNECT                              *REMOTES
**                                    **
  LISTNAME=SLUREQ1                      NAME=SEND2SLU
    TYPE=LU1RJE                         TYPE=LU1RJE
    ACSESS#=1                           MEDIA=PU
    TIME=10:00                          CONSOLE=NO
      REQUEST1 MEDIA=PU
**
*REMOTES
**
  NAME=REQUEST1
    TYPE=LU1RJE
    RMTACB=ENTPRSB
    LOGMODE=RJE3770
    CONSOLE=NO
    USERDATA=’SEND2SLU’
     MEDIA=PU
```

This operation is performed by ENTPRSA sending a batch, which contains a $$REQUEST card, to ENTPRSB. The batch sent by ENTPRSA requests the batch desired from ENTPRSB. This batch is stored on the ENTPRSA VSAM files under the Connect:Enterprise Mailbox ID REQUEST1:

```
 $$REQ ID=SEND2SLU
```

ENTPRSB sends all batches stored on its VSAM files with the ID SEND2SLU that are eligible for selection. The batches of data sent from ENTPRSB must have a $$ADD card with appropriate parameters as their first data record.

For example, the following batch is stored on the ENTPRSB VSAM files under the Mailbox ID SEND2SLU:

```
$$ADD ID=xxxxxxxx BID='yyyyyyyyyyyyy' MX=YES
DATA RECORD ONE
DATA RECORD TWO
...............
...............
END OF DATA
```

If no $$ADD card is present in the data, the batch is received in ENTPRSA as a BATCH WITHOUT $$ADD with a Mailbox ID REQUEST1.

Example 3: Connect:Enterprise (BSC)

This example shows a BSC connection with two Connect:Enterprise systems. ENTPRSA starts the Auto Connect session, requests a batch, and sends a batch to ENTPRSB. The session is started by the Auto Connect timer function and stopped when ENTPRSB has transmitted all batches for Mailbox ID SEND2A. The ODFs for ENTPRSA and ENTPRSB are shown:

```
       ENTPRSA                          ENTPRSB
*OPTIONS                             *OPTIONS
  VPF='ENTPRSA.VPF'                     VPF='ENTPRSB.VPF'
  BTAM=YES                             BTAM=YES
                                       PASSWORD=AVOCADO
**                                     SECURITY=BATCH
*CONNECT
**
  LISTNAME=CEBSC
    TYPE=BSCAD       **
    TIME=10:00
    RETRY=1
      FRESNO 07 7777777 MODE=SENDRECV
```

ENTPRSA initiates the connection with a batch containing a $$REQUEST card for the desired batch to be returned and a $$ADD for the batch to be added to ENTPRSB. The batch sent from ENTPRSA is stored on the ENTPRSA VSAM files under Mailbox ID FROMA.

```
$$REQ  ID=SEND2A  BLOCK=6
$$ADD  ID=FROMA  BID='xxxxxxxxxxxx'
DATA RECORD ONE
DATA RECORD TWO
............
............
END OF DATA
```

When the $$REQUEST command is processed by ENTPRSB, all batches stored on the ENTPRSB VSAM files with the Mailbox ID SEND2A that are eligible for selection are sent to ENTPRSA. If no $$ADD card is present, ENTPRSA receives the data as BATCH WITHOUT $$ADD, with the BSC line ID as the Mailbox ID.

Connect:Enterprise-to-Connect:Enterprise Sessions for FTP Clients and FTP Servers

Connect:Enterprise can act as both an FTP client and an FTP server. In a Connect:Enterprise-to-Connect:Enterprise session, one side is the server and one side is the client. The number of FTP client sessions available is determined by the setting in the ODF file *OPTIONS: FTP_MAX_CLIENT_THREADS parameter. The number of FTP server sessions available is determined by the setting in the ODF file *OPTIONS: FTP_MAX_SERVER_THREADS parameter.

Connect:Enterprise FTP Client to Connect:Enterprise FTP Server ODF Parameters

The following ODF parameters are in the *OPTIONS, *CONNECT, and *REMOTES sections of the ODF to control Connect:Enterprise-to-Connect:Enterprise sessions between FTP clients and FTP servers.

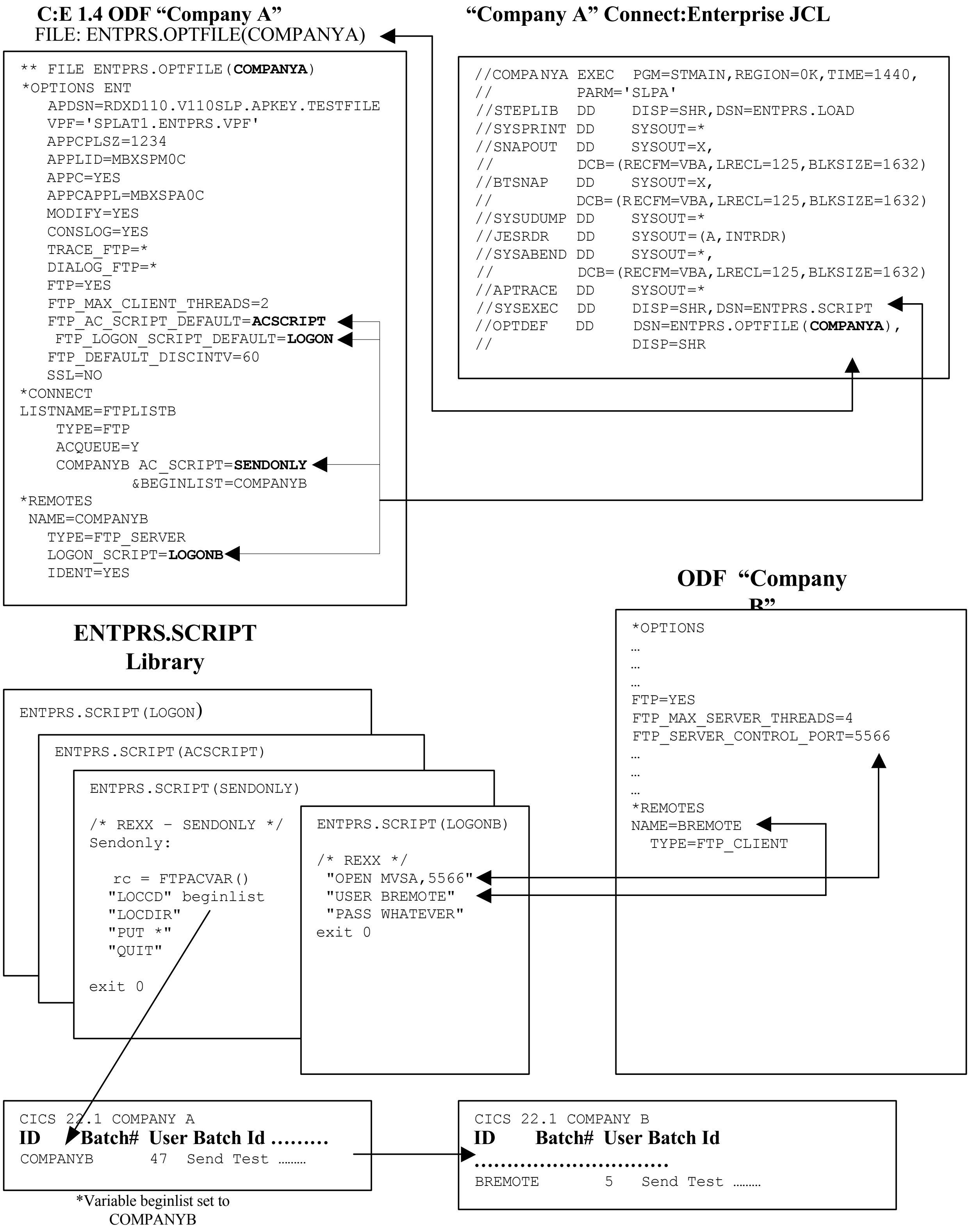
COMPANY A—Connect:Enterprise FTP Client	
Parameter	**Description**
***OPTIONS**	
FTP=YES	Specifies that Connect:Enterprise should initialize the Connect:Enterprise FTP environment.
FTP_MAX_CLIENT_THREADS=n	Specifies the maximum number of session threads available for Auto Connect sessions.
FTP_LOGON_SCRIPT_DEFAULT=xxxxxxxx	Specifies the LOGON_SCRIPT used to connect to the remote FTP server if the remote site does not specify a different script.
FTP_AC_SCRIPT_DEFAULT=xxxxxxxx	Specifies the AC_SCRIPT used if the Auto Connect list does not specify a different script.
***CONNECT**	
LISTNAME=xxxxxxxx TYPE=FTP	Creates an FTP Auto Connect list.
ftp_server_name AC_SCRIPT=	Specifies the FTP server remote sites this Auto Connect list will communicate with AC_SCRIPT identifies the REXX script that runs to send or receive files.
***REMOTES**	

NAME=ftp_server_name
TYPE=FTP_SERVER
	Identifies an FTP server. Type must be set to FTP_SERVER.

LOGON_SCRIPT	Specifies the REXX script that connects to the FTP_SERVER.

COMPANY B—Connect:Enterprise FTP Server	
Parameter	**Description**
***OPTIONS**	
FTP=YES	Specifies that Connect:Enterprise should initialize the Connect:Enterprise FTP environment.
FTP_MAX_SERVER_THREADS=n	Specifies the maximum number of session threads available for Auto Connect sessions.
FTP_SERVER_CONTROL_PORT=nnnn	Specifies the control port that FTP monitors for connection requests.
***REMOTES**	
NAME=ftp_client_name TYPE=FTP_CLIENT	Identifies an FTP client. Type must be set to FTP_CLIENT.

The following diagram maps a sample connection between a Connect:Enterprise FTP client and a Connect:Enterprise FTP server.



C:E 1.4 ODF “Company A”
FILE: ENTPRS.OPTFILE(COMPANYA)

```
** FILE ENTPRS.OPTFILE(COMPANYA)
*OPTIONS ENT
   APDSN=RDXD110.V110SLP.APKEY.TESTFILE
   VPF='SPLAT1.ENTPRS.VPF'
   APPCPLSZ=1234
   APPLID=MBXSPM0C
   APPC=YES
   APPCAPPL=MBXSPA0C
   MODIFY=YES
   CONSLOG=YES
   TRACE_FTP=*
   DIALOG_FTP=*
   FTP=YES
   FTP_MAX_CLIENT_THREADS=2
   FTP_AC_SCRIPT_DEFAULT=ACSCRIPT
    FTP_LOGON_SCRIPT_DEFAULT=LOGON
   FTP_DEFAULT_DISCINTV=60
   SSL=NO
*CONNECT
LISTNAME=FTPLISTB
    TYPE=FTP
    ACQUEUE=Y
    COMPANYB AC_SCRIPT=SENDONLY
             &BEGINLIST=COMPANYB
*REMOTES
 NAME=COMPANYB
   TYPE=FTP_SERVER
   LOGON_SCRIPT=LOGONB
   IDENT=YES
```

“Company A” Connect:Enterprise JCL

```
//COMPANYA EXEC  PGM=STMAIN,REGION=0K,TIME=1440,
//          PARM='SLPA'
//STEPLIB  DD    DISP=SHR,DSN=ENTPRS.LOAD
//SYSPRINT DD    SYSOUT=*
//SNAPOUT  DD    SYSOUT=X,
//          DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//BTSNAP   DD    SYSOUT=X,
//          DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//SYSUDUMP DD    SYSOUT=*
//JESRDR   DD    SYSOUT=(A,INTRDR)
//SYSABEND DD    SYSOUT=*,
//          DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
//APTRACE  DD    SYSOUT=*
//SYSEXEC  DD    DISP=SHR,DSN=ENTPRS.SCRIPT
//OPTDEF   DD    DSN=ENTPRS.OPTFILE(COMPANYA),
//               DISP=SHR
```

ODF “Company B”

```
*OPTIONS
…
…
…
FTP=YES
FTP_MAX_SERVER_THREADS=4
FTP_SERVER_CONTROL_PORT=5566
…
…
…
*REMOTES
NAME=BREMOTE
  TYPE=FTP_CLIENT
```

ENTPRS.SCRIPT Library

```
ENTPRS.SCRIPT(LOGON)
```

```
ENTPRS.SCRIPT(ACSCRIPT)
```

```
ENTPRS.SCRIPT(SENDONLY)

/* REXX - SENDONLY */
Sendonly:

  rc = FTPACVAR()
  "LOCCD" beginlist
  "LOCDIR"
  "PUT *"
  "QUIT"

exit 0
```

```
ENTPRS.SCRIPT(LOGONB)

/* REXX */
 "OPEN MVSA,5566"
 "USER BREMOTE"
 "PASS WHATEVER"
exit 0
```

```
CICS 22.1 COMPANY A
ID      Batch#  User Batch Id .........
COMPANYB       47  Send Test .........
```

*Variable beginlist set to COMPANYB

```
CICS 22.1 COMPANY B
ID       Batch#  User Batch Id
..............................
BREMOTE        5   Send Test .........
```

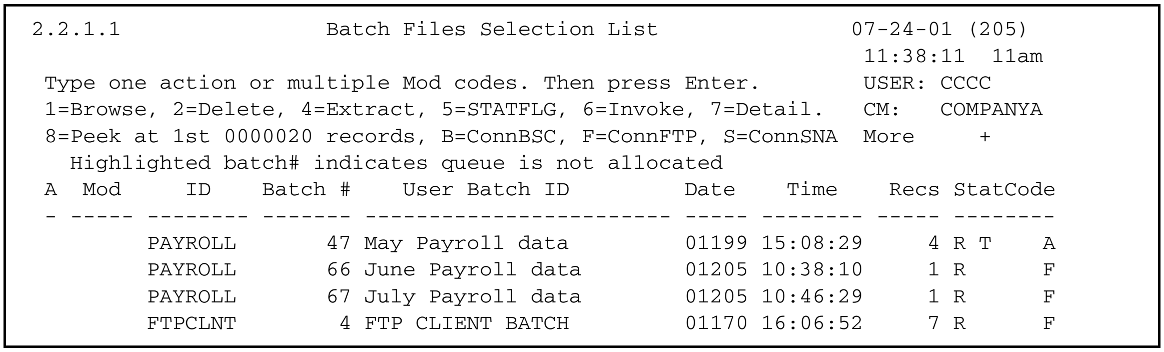
Sending a Batch from a Connect:Enterprise FTP Client to a Connect:Enterprise FTP Server

If you are using the put or mput commands to send a Connect:Enterprise batch to a remote Connect:Enterprise system, you use the cd, loccd, locdir commands to select the Mailbox ID to transfer from and to.

The loccd and locdir commands refer to the local Connect:Enterprise client system.

The cd command refers to the remote Connect:Enterprise server system.

For example, list the following batches for Company A using the CICS interface:



```
2.2.1.1                Batch Files Selection List              07-24-01 (205)
                                                                11:38:11  11am
 Type one action or multiple Mod codes. Then press Enter.       USER: CCCC
 1=Browse, 2=Delete, 4=Extract, 5=STATFLG, 6=Invoke, 7=Detail.   CM:   COMPANYA
 8=Peek at 1st 0000020 records, B=ConnBSC, F=ConnFTP, S=ConnSNA  More     +
   Highlighted batch# indicates queue is not allocated
 A  Mod     ID    Batch #    User Batch ID         Date    Time    Recs StatCode
 - ----- -------- ------- ----------------------- ----- -------- ----- --------
         PAYROLL       47 May Payroll data         01199 15:08:29     4 R T    A
         PAYROLL       66 June Payroll data        01205 10:38:10     1 R      F
         PAYROLL       67 July Payroll data        01205 10:46:29     1 R      F
         FTPCLNT        4 FTP CLIENT BATCH         01170 16:06:52     7 R      F
```

To send the PAYROLL batches that have not already been marked transmitted (batch #66 and #67), your AC script should have the following command before the PUT * command:

```
“LOCCD” PAYROLL
“LOCDIR”
“PUT *”
```

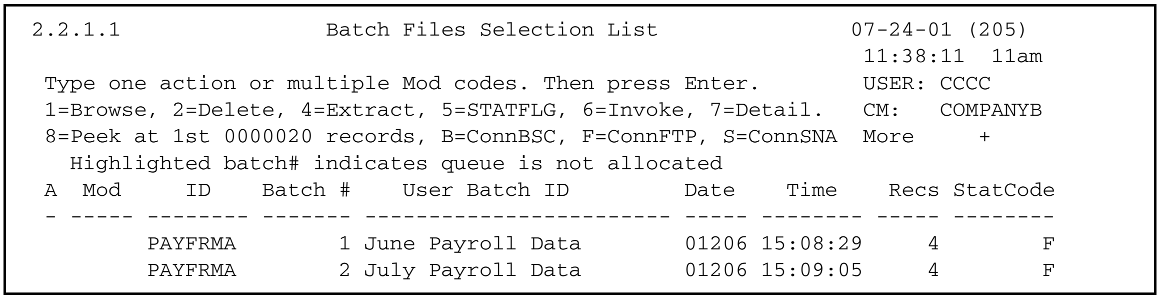
This selects PAYROLL batches #66 and #67 for transmission (assuming DIR_FILTER, FTIME, TTIME, ORIGIN and ONEBATCH site specifications do not exclude batch #66 or #67).

Specifying the Connect:Enterprise Mailbox ID on an FTP Remote Server

To specify a Mailbox ID to assign to the transmitted batches in the receiving Connect:Enterprise system, use the cd command:

```
“LOCCD” PAYROLL
“LOCDIR”
“CD PAYFRMA”
“PUT *”
```

PAYROLL batches #66 and #67 are transferred to the receiving Connect:Enterprise system and stored in Mailbox PAYFRMA.



```
2.2.1.1               Batch Files Selection List              07-24-01 (205)
                                                               11:38:11  11am
 Type one action or multiple Mod codes. Then press Enter.      USER: CCCC
 1=Browse, 2=Delete, 4=Extract, 5=STATFLG, 6=Invoke, 7=Detail.  CM:   COMPANYB
 8=Peek at 1st 0000020 records, B=ConnBSC, F=ConnFTP, S=ConnSNA  More     +
   Highlighted batch# indicates queue is not allocated
 A  Mod     ID    Batch #    User Batch ID         Date    Time    Recs StatCode
 - ----- -------- ------- ----------------------- ----- -------- ----- --------
       PAYFRMA        1 June Payroll Data        01206 15:08:29     4        F
       PAYFRMA        2 July Payroll Data        01206 15:09:05     4        F
```

Gentran Server Communications Module for OS/400 (BSC) with Connect:Enterprise for z/OS

Gentran Server Communications Module for OS/400 uses a scripting methodology to describe the direction and flow of a communication session. In particular for BSC sessions, Gentran Server Communications Module for OS/400 performs the function of the MODE= parameter in the Auto Connect list ($$CONNECT and *CONNECT commands).

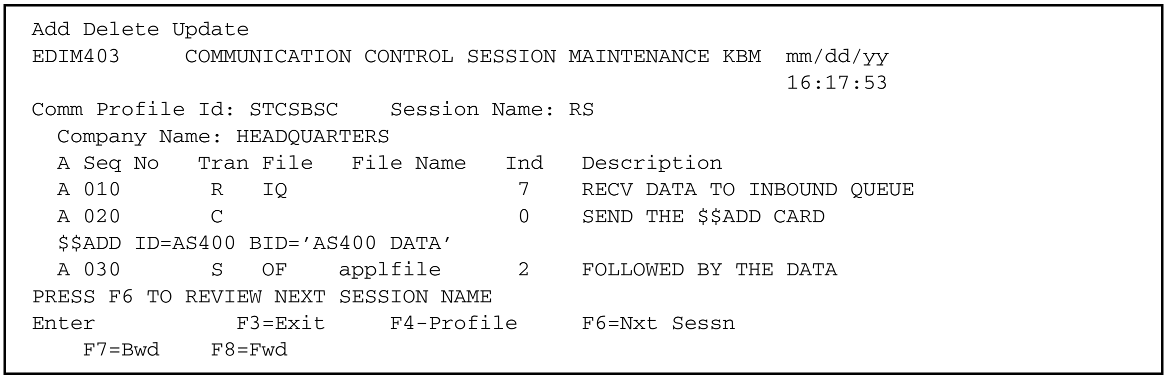
Initiating the Connection from Gentran Server Communications Module for OS/400

If Gentran Server Communications Module for OS/400 initiates the connection to Connect:Enterprise for z/OS, the script must always specify a SEND operation first to either transmit data or to send a $$REQUEST. Then it can specify a receive if data is expected from Connect:Enterprise.

Initiating the Connection from Connect:Enterprise for z/OS

If Connect:Enterprise for z/OS initiates the session, it must first send a batch to the AS/400. You must specify SENDONLY or SENDRECV for the MODE parameter in the Auto Connect list to ensure that the AS/400 first receives data when it is called.

The following Gentran Server Communications Module for OS/400 screen contains a sample script used when Connect:Enterprise for z/OS initiates the session. With this script, Gentran Server Communications Module for OS/400 receives a batch (or batches) and then turns around and transmits a $$ADD followed by an AS/400 file.



```
Add Delete Update
EDIM403     COMMUNICATION CONTROL SESSION MAINTENANCE KBM  mm/dd/yy
                                                           16:17:53
Comm Profile Id: STCSBSC    Session Name: RS
  Company Name: HEADQUARTERS
  A Seq No   Tran File   File Name   Ind   Description
  A 010       R   IQ                  7    RECV DATA TO INBOUND QUEUE
  A 020       C                       0    SEND THE $$ADD CARD
  $$ADD ID=AS400 BID='AS400 DATA'
  A 030       S   OF    applfile      2    FOLLOWED BY THE DATA
PRESS F6 TO REVIEW NEXT SESSION NAME
Enter           F3=Exit     F4-Profile     F6=Nxt Sessn
    F7=Bwd    F8=Fwd
```

The following Connect:Enterprise ODF file contains the Auto Connect list that initiates a session to Gentran Server Communications Module for OS/400 configured with the preceding script. Notice that compression and truncation are specified in the list. These capabilities are valid when transmitting to Gentran Server Communications Module for OS/400.

```
*OPTIONS
  BTAM=YES
  UA=UA412
  MODIFY=YES
**
*CONNECT
**
  LISTNAME=AS400
    TYPE=BSCAD
    AS400 07 5551212 M=SR CMP=Y TRUNC=Y
```

Connect:Enterprise SNA Transmissions

Gentran Server Communications Module for OS/400 is capable of conducting SNA file transfer sessions with Connect:Enterprise for z/OS in both remote-initiated and host-initiated modes.

Setting Up Gentran Server Communications Module for OS/400

See the Gentran Server AS/400 documentation for more information about linking it with Connect:Enterprise for z/OS. You need to supply the AS/400 site with the following items to properly configure Gentran Server Communications Module for OS/400:

- LU local address (in hex)
- Connect:Mailbox application ID
- Mainframe Exchange ID
- Mainframe SSCP ID
- Station address (in hex)
- Maximum RU size

Setting Up Connect:Enterprise for z/OS

To set up the Connect:Enterprise for z/OS on the mainframe, make changes to the VTAM mode and USS tables, NCP definitions, and the Connect:Enterprise for z/OS ODF.

VTAM Mode Table

Add the following entry to your VTAM mode table:

```
                                                                   72
STCS400  MODEENT LOGMODE=STCS400,                                   X
               FMPROF=X’03’,TSPROF=X’03’,                           X
               PRIPROT=X’B1’,SECPROT=X’A3’,COMPROT=’7080’           X
               RUSIZES=X’8686’,                                     X
               PSERVIC=X’01102000F100E00000010040’
MODEEND
```

The RUSIZE specified must be compatible with the BLKSIZE parameter in the Connect:Enterprise ODF. The compatible values for common specifications are shown in the following table:

RUSIZE	BLKSIZE
8585	256
8686	512
8989	4096

This example uses a BLKSIZE of 512 bytes, so RUSIZE is specified 8686.

VTAM USS Table

Create a VTAM USS Table entry to allow Gentran Server Communications Module for OS/400 to initiate sessions.

The following is a sample USS table entry logon to the mainframe as remote name:

```
LOGON   USSCMD  CMD=LOGON,REP=LOGON,FORMAT=PL1
        USSPARM PARM=APPLID,DEFAULT=ENTPRS01
        USSPARM PARM=DATA,DEFAULT=remote name
        USSMSG MSG=10,SUPP=ALWAYS
```

The remote name specified in the USS table entry must match a valid remote name defined in the Connect:Enterprise for z/OS ODF *REMOTES section.

NCP LU Definition

Add an LU definition to your NCP, for each concurrent session with Gentran Server Communications Module for OS/400. The following is a sample NCP LU definition:

```
                                                                   72
luname  LU  LOCADDR=localaddr,                                      X
        DLOGMOD=STCS400                                             X
        USSTAB=usstable
```

Connect:Enterprise ODF

If sessions are host-initiated, set up remote sites specifications and Auto Connect lists in the ODF as shown in the following example:

```
*OPTIONS
  VTAM=YES
  VPF=’ENTPRS.VPF’
  APPLID=ENTPRS01
  MODIFY=YES
**
*REMOTES
**
  NAME=SNARMT01
    CONSOLE=YES
    COMPRESS=NO
    MEDIA=CN
    BLKSIZE=512
**
*CONNECT
**
  LISTNAME=SNALIST1
    TYPE=LU1RJE
    ACSESS#=1
    DISCINTV=30
    RETRY=2
    MAXRMT#0=1
      SNARMT01 MEDIA=CN
```

Connect:Tracs for MVS and VSE (BSC) with Connect:Enterprise for z/OS

When Connect:Tracs for MVS and VSE (BSC) are used with Connect:Enterprise, Connect:Enterprise considers Connect:Tracs a remote BSC terminal, even if it runs on the same host as Connect:Enterprise. The terms *remote site* or *remote ID* still apply to Connect:Tracs. All data formats described for Connect:Enterprise must be used by Connect:Tracs.

Connect:Tracs needs access to a communications port and must contact Connect:Enterprise using one of the allocated Connect:Enterprise communication lines, even if it runs on the same host as Connect:Enterprise. Connect:Tracs can also run in a remote host computer and communicate with Connect:Enterprise.

Connect:Tracs operates in one of four modes. If Connect:Enterprise initiates the connection to Connect:Tracs through an Auto Connect session, any of the four Connect:Tracs modes can be used. However, the Connect:Tracs mode must be compatible with the Connect:Enterprise mode.

If Connect:Tracs initiates the connection to Connect:Enterprise, one of the following modes can be used:

- Connect:Tracs 3780/2780 emulation mode
- Connect:Tracs send-only mode

The following sections describe the two modes for Connect:Tracs-initiated connections to Connect:Enterprise.

Connect:Tracs 3780/2780 Emulation Mode

The 3780/2780 emulation mode first sends one or more batches to Connect:Enterprise, then receives one or more batches from Connect:Enterprise.

The following sample data stream sends a data file from Connect:Tracs to Connect:Enterprise, then receives all batches at the host queued for the Mailbox ID (assume TRACS as the ID). Use Connect:Tracs generated for 3780/2780 emulation mode.

```
$$REQUEST ID=TRACS BLOCK=6
$$ADD ID=TRACSA BATCHID='*MEMO TO ALL BRANCHES *'
.
.
data records
.
.
```

The Connect:Tracs receive file is used for the data received due to the $$REQUEST in the input data. You must customize a Connect:Tracs transmission module for use in 3780/2780 emulation mode.

The available Connect:Tracs macro keywords that customize a Connect:Tracs transmission module follow, with special notes on those options used with Connect:Enterprise.

Connect:Tracs Keywords for 3780/2780 Emulation Mode

The following is a list of Connect:Tracs keywords for use in the 3780/2780 emulation mode:

Keyword	Description
COMPRES	Connect:Enterprise can process any of the valid COMPRES options.
DEVICE	Connect:Enterprise can process any of the valid DEVICE options.
DFTLST	Connect:Enterprise can process any of the valid DFTLST options.
INPBLK	Connect:Enterprise can process any number of blocked input records as long as the total length does not exceed 4,096 characters. The default value, 5, is valid for the most common record size, which is 80 characters.
INPSIZE	Connect:Enterprise can send records up to 4,094 characters in length. Connect:Enterprise can send records up to 4,093 characters in length if the data is transparent.
LOGOFF	NO must be specified. This is the default value.

Keyword	Description
MULTRCV	Use the default value, NO, unless your input data stream contains $$REQUEST or $$DIRECTORY inquiries, which can cause multiple files to be sent from Connect:Enterprise to Connect:Tracs. Connect:Enterprise uses an EOT character to separate files when multiple batches are transmitted to Connect:Tracs. If NO is used, Connect:Tracs does not receive all the data from Connect:Enterprise.
OS	DOS must be specified for VSE. MVS must be specified for z/OS.
RCVBUFF	The value must not exceed 4096. The default value is 514.
RECSEP	Connect:Enterprise can process any of the valid RECSEP options.
SNDBUFF	The value must not exceed 4096. The default value is 512.
SWLINE	Connect:Enterprise can process any of the valid SWLINE options.
TRNPAR	Connect:Enterprise can process any of the valid TRNPAR options.
WAITC	NO must be specified. This is the default value.

When using Connect:Tracs with Connect:Enterprise, always use the default values for all uncommon keywords, with two possible exceptions:

- Connect:Enterprise forces TRUNC=YES to drop trailing blanks from data sent to and received from Connect:Tracs.
- Connect:Enterprise enables the IDVER parameter with Connect:Tracs.

Connect:Tracs Send-Only Mode

The send-only mode sends one or more batches to Connect:Enterprise, then disconnects.

The following sample data stream is used as the Connect:Tracs send file to add two batches to the Connect:Enterprise VSAM batch files. These batches would then be available for transmission to other remote sites in the network. In this example, the Mailbox ID assigned to the batches is the ID of the remote sites to later receive the batches, rather than the IDs for Connect:Tracs.

```
$$ADD ID=HOUSTON BATCHID=’6/19 ACCTS RECEIVABLE’ XMIT=YES
.
.
data records
.
$$ADD ID=DALLAS BATCHID=’6/19 ACCTS PAYABLE’ XMIT=YES
.
data records.
.
```

Since multiple $$ADD records are used in a single Connect:Tracs send file, the Connect:Tracs module generated should specify INPBLK=1.

You must customize a Connect:Tracs transmission module for use in Send Only mode.

The Connect:Tracs macro keywords that customize a Connect:Tracs transmission module are listed in the following table, with special notes on those options used with Connect:Enterprise.

Connect:Tracs Keywords for Send-Only Mode

The following is a list of Connect:Tracs keywords for use in the send only mode:

Keyword	Description
DEVICE	HOST is required by Connect:Tracs for send-only mode.
DFTLST	Connect:Enterprise can process any of the valid DFTLST options.
INPBLK	Connect:Enterprise can process any number of blocked input records as long as the total length does not exceed 4,096 characters. The default value, 5, is valid for the most common record size, which is 80 characters. If the input records sent to Connect:Enterprise contain multiple $$ADD records to separate the data into batches, you must specify INPBLK=1.
INPSIZE	Connect:Enterprise can receive input records up to 4096 characters.
OS	DOS must be specified for VSE. MVS must be specified for z/OS.
SNDBUFF	The value must not exceed 4096. The default value is 512.
SNDONLY	YES is required by Connect:Tracs for send-only mode.
SWLINE	Connect:Enterprise can process any of the valid SWLINE options.
TRNPAR	Connect:Enterprise can process any of the valid TRNPAR options.

When using Connect:Tracs with Connect:Enterprise, always use the default values for all uncommon keywords, with two possible exceptions. Connect:Enterprise uses TRUNC=YES to drop trailing blanks from data received from Connect:Tracs. Connect:Enterprise also enables the IDVER parameter with Connect:Tracs.

Connect:Enterprise Gateway with Connect:Enterprise for z/OS

When configuring Connect:Enterprise remote definitions for Connect:Enterprise Gateway (Gateway), it is important to separate the processes that are used for remote- and host-initiated connections. Once separated, they can each be handled properly. This section provides an overview on how to set up Connect:Enterprise in both instances and how Connect:Enterprise definitions correspond to Gateway definitions.

Defining the Switched Major Node for Connect:Enterprise Gateway

Before creating any Connect:Enterprise definitions for a Gateway remote site, you must create the VTAM definition that connects to Gateway. Following is a sample switched major node definition for a token ring connection. Your definition will be similar to this sample, but not identical.

```
*********************************************************************** 00010000
*                      SWITCHED SNA DEFINITION                        * 00020000
*                                                                     * 00030000
* CHANGE LOG:                                                         * 00040000
* 01/31/01  WSTSBWM   CREATE FOR Gateway Test ENV QA Team DE & JV     * 00050000
*********************************************************************** 00060000
M1T2411  VBUILD TYPE=SWNET                                              00070000
*                                                                       00080000
M1T2411P PU    ADDR=01,                                                X00090000
               IDBLK=071,                                              X00100000
               IDNUM=02411,                                            X00110000
               ISTATUS=ACTIVE,                                         X00120000
               DISCNT=NO,                                              X00130000
               SSCPFM=USSSCS,                                          X00140000
               MODETAB=DALLMTAB,                                       X00150000
               PUTYPE=2,                                               X00160000
               VPACING=7,                                              X00180000
               USSTAB=ISTINCDT,                                        X00190000
               DLOGMOD=RJE3770D                                         00200000
*                                                                       00210000
*                                                                       00220000
M1241102 LU    LOCADDR=2                                                00230000
M1241103 LU    LOCADDR=3                                                00240000
M1241104 LU    LOCADDR=4                                                00250000
M1241105 LU    LOCADDR=5                                                00260000
M1241106 LU    LOCADDR=6                                                00270000
M1241107 LU    LOCADDR=7                                                00280000
M1241108 LU    LOCADDR=8                                                00290000
M1241109 LU    LOCADDR=9                                                00300000
M1241110 LU    LOCADDR=10                                               00310000
M1241111 LU    LOCADDR=11                                               00320000
M1241112 LU    LOCADDR=12                                               00330000
M1241113 LU    LOCADDR=13                                               00340000
M1241114 LU    LOCADDR=14                                               00350000
M1241115 LU    LOCADDR=15                                               00360000
M1241116 LU    LOCADDR=16                                               00370000
M1241117 LU    LOCADDR=17                                               00380000
M1241118 LU    LOCADDR=18                                               00390000
M1241119 LU    LOCADDR=19                                               00400000
M1241120 LU    LOCADDR=20                                               00410000
M1241121 LU    LOCADDR=21                                               00420000
M1241122 LU    LOCADDR=22                                               00430000
M1241123 LU    LOCADDR=23                                               00440000
M1241124 LU    LOCADDR=24                                               00450000
M1241125 LU    LOCADDR=25                                               00460000
M1241126 LU    LOCADDR=26                                               00470000
M1241127 LU    LOCADDR=27                                               00480000
M1241128 LU    LOCADDR=28                                               00490000
M1241129 LU    LOCADDR=29                                               00500000
M1241130 LU    LOCADDR=30                                               00510000
M1241131 LU    LOCADDR=31                                               00520000
M1241132 LU    LOCADDR=32                                               00530000
M1241133 LU    LOCADDR=33                                               00540000
M1241134 LU    LOCADDR=34                                               00550000
M1241135 LU    LOCADDR=35                                               00560000
M1241136 LU    LOCADDR=36                                               00570000
M1241137 LU    LOCADDR=37                                               00580000
M1241138 LU    LOCADDR=38                                               00590000
M1241139 LU    LOCADDR=39                                               00600000
M1241140 LU    LOCADDR=40                                               00610000
M1241141 LU    LOCADDR=41                                               00620000
M1241142 LU    LOCADDR=42                                               00630000
M1241143 LU    LOCADDR=43                                               00640000
M1241144 LU    LOCADDR=44                                               00650000
M1241145 LU    LOCADDR=45                                               00660000
M1241146 LU    LOCADDR=46                                               00670000
M1241147 LU    LOCADDR=47                                               00680000
M1241148 LU    LOCADDR=48                                               00690000
M1241149 LU    LOCADDR=49                                               00700000
M1241150 LU    LOCADDR=50                                               00710000
M1241151 LU    LOCADDR=51                                               00720000
M1241152 LU    LOCADDR=52                                               00730000
M1241153 LU    LOCADDR=53                                               00740000
```

Verify the MODETAB name is the same as the one that was specified when installing Microsoft Host Integration Server (CM/2) and Gateway. Make note of the maximum RU size that can be sent. Any attempt to send larger data blocks results in errors. Other information from this definition is required during the installation and configuration of both Host Integration Server and Gateway.

When defining the LU names for the switched major node, specify enough names for both inbound and outbound connections.

Always refer to the Connect:Enterprise Gateway documentation for the correct VTAM definition requirements.

Creating Remote Definitions

Following is a sample remote definition for a Connect:Enterprise Gateway remote site. This definition controls processing whenever a remote site connects with Connect:Enterprise (through Gateway).

```
NAME=SPC1021              INBOUND FROM SPC1
   SC=SPC
   LUNAME=SPC1021
   LOGMODE=RJE3770D
   TYPE=LU1RJE
   COMPRESS=NO
   CONSOLE=YES
   MEDIA=PU
   DISCINTV=60
```

The following table describes the parameters:

Parameter	Description
SC	Code SC=SPC for any connection to Gateway.
LUNAME	Specifies one of the inbound LU names created in the VTAM definition.
COMPRESS	Indicates compression is done. If not set properly, a warning message (CMB184I) is issued during session startup.
CONSOLE	CONSOLE=YES must always be specified for any Connect:Enterprise Gateway remote sites that may start remote connects with Connect:Enterprise. If the remote definition is created only for Auto Connect sessions, specify CONSOLE=NO if the remote user does not want to receive error messages from Connect:Enterprise and store them as files.
MEDIA	MEDIA=PU must be specified for Connect:Enterprise Gateway remote sites.
DISCINTV	The DISCINTV= value has a relationship to several timer values that are specified in Connect:Enterprise Gateway. For remote connections, these are the Connection Time-out value in the LUA account and the Inactivity Time-out value in the RPF account. The relationship of the three values should be: RPF Inactivity Time-out > Connect:Enterprise DISCINTV > LUA Connection Time-out For example, if the RPF Inactivity Time-out value in the remote definition is 70, the Connect:Enterprise DISCINTV value should be 60, and the LUA Connection Time-out value should be 50.

The size of data blocks sent to Connect:Enterprise from Connect:Enterprise Gateway is usually set to the same size as the maximum RU. Lower this value if you have a high number of concurrent sessions, or if you are running Connect:Enterprise Gateway on a PC with a small amount of RAM or disk space. By reducing the block size, Connect:Enterprise Gateway sends data to Connect:Enterprise more frequently, thereby reducing the chance of a time-out. By reducing the maximum block size value, the number of I/Os increases.

Creating Auto Connect Session Definitions

Following is an Auto Connect session definition that connects with a Connect:Enterprise Gateway remote site.

```
LISTNAME=LSPC1               <== AC BY REMOTE NAME AND LISTNAME
   TYPE=LU1RJE
   DISCINTV=70
   SPC1001  IDLIST=SPCXDATA
```

The DISCINTV= value overrides the value specified in the remote definition. This may be necessary to allow additional time for Connect:Enterprise Gateway to make the connection to the remote.

Following is the remote definition that is pointed to by the Auto Connect session definition:

```
NAME=SPC1001              OUTBOUND TO SPC1
   SC=SPC
   POOL=POOLSPC1
   LOGMODE=RJE3770D
   TYPE=LU1RJE
   COMPRESS=NO
   CONSOLE=YES
   MEDIA=PU
   DISCINTV=60
   BLKSIZE=4096
```

The following table describes the parameters:

Parameter	Description
SC	Code SC=SPC for any connection to Connect:Enterprise Gateway.

POOL	The POOL parameter specifies the list of outbound LU names created in the VTAM definition. Pooling is useful for Auto Connect sessions when more remote users are defined to Connect:Enterprise Gateway than LU Names are defined to VTAM. By using a pool of LU names, Connect:Enterprise has greater connection success with Connect:Enterprise Gateway.
	The following is a sample Pool definition:
	`NAME=POOLSPC1`
	`LU=SPC1001,SPC1002,SPC1003,SPC1004,SPC1005,SPC1006,SPC1007,SPC1008`
	`LU=SPC1009,SPC1010,SPC1011,SPC1012,SPC1013,SPC1014,SPC1015,SPC1016`

COMPRESS	Indicates compression is done. If not set properly, a warning message (CMB184I) is issued during session startup.

Parameter	Description
CONSOLE	CONSOLE=YES must be specified for any Connect:Enterprise Gateway remote sites that may start remote connects with Connect:Enterprise. If the remote definition is created only for Auto Connect sessions, specify CONSOLE=NO if the remote user does not want to receive error messages from Connect:Enterprise and store them as files.
MEDIA	MEDIA=PU should be specified for Connect:Enterprise Gateway remote sites.
DISCINTV	The DISCINTV= value has a relationship to several timer values specified in Connect:Enterprise Gateway. For remote connections, these are the Connection Time-out value in the LUA account and the Inactivity Time-out value in the RPF account. The relationship of the three values should be: RPF Inactivity Time-out > Connect:Enterprise DISCINTV > LUA Connection Time-out For example, if the RPF Inactivity Time-out value is 70, the Connect:Enterprise DISCINTV value in the remote definition should be 60, and the LUA Connection Time-out value should be 50. Note that the DISCINTV= value in the Auto Connect session definition overrides the value specified in the remote definition. Set the time-out values in Connect:Enterprise Gateway based on that value.

Output Media for Connect:Enterprise Gateway Remote Sites

All batches received from Connect:Enterprise Gateway have output media of PUNCH (PU). Assign an output media of PU to any batches that Connect:Enterprise sends to Connect:Enterprise Gateway by specifying MEDIA=PU on the Connect:Enterprise Gateway Remote Definition.

By default, any batches that Connect:Enterprise Gateway sends to 3770 remote sites use an output media of PU.

Chapter 12

Running Connect:Enterprise

This chapter describes the following topics:

- Starting the VSAM file server
- Starting Connect:Enterprise
- Shutting down Connect:Enterprise

Starting the VSAM File Server

You must first start the VSAM file server to bring up Connect:Enterprise. To start the VSAM file server, issue the following command from the system console:

```
S procname
```

where *procname* is the name of the VSAM file server startup process created during the Connect:Enterprise installation.

The following message is displayed when the VSAM file server starts:

```
BTB002I : VSAM server initialization complete.
```

Starting Connect:Enterprise

After you start the VSAM file server, issue the following command from the system to start Connect:Enterprise:

```
S procname
```

where *procname* is the name of the Connect:Enterprise online system. See the *Connect:Enterprise for z/OS for z/OS Installation Guide* for more information about creating the Connect:Enterprise startup task.

Shutting Down Connect:Enterprise

Perform the following steps to shut down Connect:Enterprise and the VSAM file server.

Caution: Always shut down Connect:Enterprise and all offline utilities before shutting down the VSAM file server.

1. Enter the following command at the system console to shut down Connect:Enterprise and all offline utilities. In this example, *cename* represents the name of the Connect:Enterprise system or the started task that runs Connect:Enterprise.

   ```
   F cename,$$SHUTDOWN
   ```

2. Enter the following command at the system console to shut down the VSAM file server, where *procname* is the VSAM file server process name:

   ```
   F procname,$$SER STOP
   ```

Chapter 13

File Maintenance

This chapter describes the following topics:

- File maintenance overview
- VBQ file maintenance
- VLF file maintenance
- VPF file maintenance
- VCF file maintenance
- VCF Alternate Index maintenance
- Batch number maintenance

Note: Use the VERIFY Utility to produce a report listing inconsistencies between VCF, VPF, and VBQ files. This utility also allows you to validate and, if necessary, resync VSAM VPF, VCF, and VBQ files. See the *Offline Utilities* chapter of *Connect:Enterprise for z/OS User's Guide* for more information.

File Maintenance Overview

To maintain a consistent level of performance for Connect:Enterprise, as batches are added to and erased from the VSAM batch queues, you must recover the VSAM space and eliminate any CI/CA splits.

Running the ERASE utility (see the *Offline Utilities* chapter of *Connect:Enterprise for z/OS User's Guide*) frees space within the VSAM batch files. However, it does not ensure that Connect:Enterprise can reuse the space, due to VSAM and KSDS cluster consideration. So, you must regularly monitor VSAM batch file physical status and perform maintenance when any of the following occur:

- CI splits
- CA splits

- Multiple extents
- Slow VSAM performance
- High-used RBA is close to the high-allocated RBA (high-used RBA never decreases, even if records are deleted)

VBQ File Maintenance

If you defined the VBQROTAT and the VBQPCT parameters in the ODF *OPTIONS section, Connect:Enterprise monitors the current collection VBQ space utilization. Once the current collection VBQ used space reaches the threshold defined in VBQPCT, or enters secondary extents, batch collection automatically rotates to the next eligible VBQ. If the VBQPCT threshold is reached while a batch is being collected, collection does not rotate to the next VBQ until the next online batch begins collection. This ensures that all records for a batch are on the same VBQ file.

Only online Connect:Enterprise performs automatic VBQ rotation; offline utility processing does not perform it. (However, you can manually change the current collection VBQ by using the $$ALLOC console command.)

If you do not use the VBQROTAT and the VBQPCT parameters, monitor space usage with the $$SPACE or $$SPACEX console command. Manually change the current collection VBQ with the $$ALLOC console command.

To reclaim VSAM space and reduce CI or CA splits, deallocate a VBQ file that is not the current collection VBQ. Be careful of when and which VBQ files you deallocate because any batches that reside on a deallocated VBQ file are not available for transmission to a remote site.

You can use the MOVE utility to move any remaining batches from the VBQ prior to doing maintenance. This keeps all batches available for transmission and makes maintenance easier.

Preventing Offline Utility Processing Against a Deallocated VBQ

You can prevent the offline ADD, EXTRACT, MOVE, and ERASE utilities from processing against deallocated VBQs. When an operator runs one of these utilities against a deallocated VBQ, Connect:Enterprise returns a warning message indicating that the VBQ is offline. The default return code value of the message is RC=4 (Processing Continues). However, you can change the return code to a higher value to stop this utility (and others if desired) from executing. See the *Offline Utilities* chapter of *Connect:Enterprise for z/OS User’s Guide* for more information about return code values.

Reclaiming Space from a VBQ with Data

Use the following procedure to reclaim unused space and remove CA/CI splits on a VBQ file that is not the current collection file, when you want to retain the file’s data:

1. Verify that the ERASE utility has been run to erase any batches that are no longer needed.
2. Verify that no offline utility jobs are executing that require access to the VBQ.

3. Deallocate the VBQ with the $$DALLOC console command, or use the Deallocate File Request CICS or ISPF panel.
4. Perform the following using IDCAMS:
    a. Use the REPRO command to back up your VBQ file batch data.
    b. Use the DELETE command to delete the VBQ file.
    c. Use the DEFINE command to define the VBQ file with the same data set name as previously used. Do not change this name.
    d. Use the REPRO command to copy the data from the backup VBQ file into the newly defined VBQ file.
5. Allocate the VBQ file to Connect:Enterprise with the $$ALLOC console command or use the Allocate File Request CICS or ISPF panel.

    You can make it the current collection file by appending the C parameter to the $$ALLOC command.

The VBQ file is now available to Connect:Enterprise with reclaimed space and no CI/CA splits.

Note: For best results, back up all VSAM files to a sequential RECFM=VB to prevent losing the dummy record during the IDCAMS REPRO backup procedure.

Reclaiming Space from a VBQ without Data

Use the following procedure to reclaim unused space and remove CA/CI splits on a VBQ file that is not the current collection file and that contains no batch data:

1. Verify that no offline utility jobs are executing that require access to the VBQ.
2. Deallocate the VBQ with the $$DALLOC console command or use the Deallocate File Request CICS or ISPF panel.
3. Perform the following using IDCAMS:
    a. Use the REPRO command to back up your VBQ file control information.
    b. Use the DELETE command to delete the VBQ file.
    c. Use the DEFINE command to define the VBQ file with the same data set name as previously used. Do not change this name.
    d. Use the REPRO command to copy the data from the backup VBQ file into the newly defined VBQ file.
4. Allocate the VBQ file to Connect:Enterprise with the $$ALLOC console command or use the Allocate File Request CICS or ISPF panel.

    You can make it the current collection file by appending the C parameter to the $$ALLOC command.

The VBQ file is now available to Connect:Enterprise with reclaimed space and no CI/CA splits.

VLF File Maintenance

If you defined the VLFROTAT and the VLFPCT parameters in the ODF *OPTIONS section, Connect:Enterprise monitors the current VLF log file space utilization. Once the current VLF used space reaches the threshold defined in VBQPCT, or enters secondary extents, Connect:Enterprise switches to the next eligible VLF.

Only online Connect:Enterprise performs automatic VLF rotation; offline utility processing does not perform it. However, you can manually change the current VLF by using the $$ALLOC console command.

If you do not use the VLFROTAT and the VLFPCT parameters, monitor space usage with the $$SPACE or $$SPACEX console command. Manually change the current VLF with the $$ALLOC console command.

You can use the same procedures to maintain the VLFnn log files that you want to empty or reuse.

Reclaiming Space from a VLF with Data

Use the following procedure to reclaim unused space or remove CA or CI splits on a VLF file that is not the current collection file, when you want to retain the file's data:

1. Verify that no offline utility jobs are accessing the VLF and that no transmissions or collections have opened log records on the VLF.
2. Deallocate the VLF with the $$DALLOC console command, or use the Deallocate File Request CICS or ISPF panel.
3. Perform the following using IDCAMS:
   a. Use the REPRO command to back up your VLF file batch data.
   b. Use the DELETE command to delete the VLF file.
   c. Use the DEFINE command to define the VLF file with the same data set name as previously used. Do not change this name.
   d. Use the REPRO command to copy the data from the backup VLF file into the newly defined VLF file.
4. Allocate the VLF file to Connect:Enterprise with the $$ALLOC console command or use the Allocate File Request CICS or ISPF panel.

   You can make it the current collection file by appending the C parameter to the $$ALLOC command.

The VLF file is now available to Connect:Enterprise with reclaimed space and no CI/CA splits.

Reclaiming Space from a VLF without Data

Use the following procedure to reclaim unused space or remove CA or CI splits on a VLF file that is not the current collection file and that contains no batch data:

1. Verify that no offline utility jobs are executing that require access to the VLF.

2. Deallocate the VLF with the $$DALLOC console command or use the Deallocate File Request CICS or ISPF panel.
3. Perform the following using IDCAMS:
   a. Use the REPRO command to back up your VLF file control information.
   b. Use the DELETE command to delete the VLF file.
   c. Use the DEFINE command to define the VLF file with the same data set name as previously used. Do not change this name.
   d. Use the REPRO command to copy the data from the backup VLF file into the newly defined VLF file.
4. Allocate the VLF file to Connect:Enterprise with the $$ALLOC console command or use the Allocate File Request CICS or ISPF panel.

   You can make it the current collection file by appending the C parameter to the $$ALLOC command.

The VLF file is now available to Connect:Enterprise with reclaimed space and no CI/CA splits.

VPF File Maintenance

The VPF file is also susceptible to CI or CA splits that could degrade performance if left unattended. You cannot deallocate the VPF. To perform file maintenance on this file, you must first shut down Connect:Enterprise.

After shutting down Connect:Enterprise, stop the VSAM File Server.

Using IDCAMS, perform the following:

1. Back up your VPF file with REPRO in order to save any valid batches you want to retain.
2. Use the DELETE command to delete the VPF file.
3. Use the DEFINE command to define the VPF file with the same data set name as previously used. Do not change this name.
4. REPRO the data back into the VPF file using as input the backup you created previously.

You can now start the VSAM file server and Connect:Enterprise.

VCF File Maintenance

The VCF file is not susceptible to CI or CA splits because all possible batch records are preallocated and initialized during PURGE processing. No file maintenance is required for this file.

VCF Alternate Index Maintenance

If you implemented the VCF Alternate Index file feature when initializing the VSAM batch files, STOUTL uses the index information as a means to more quickly access files when running offline utility programs. To verify that the alternate keys in VCF records are being properly maintained, run the STUTAVIX job, which is provided as an example member.

The STUTAVIX job performs the following functions:

- Copies the VCF to a sequential file
- Executes the VCF alternate index key validation utility
- Writes all incorrect alternate key values including a copy of all incorrect VCF to an output file

Note: Even if no VCF alternate index errors are detected, periodic maintenance is still required. Example member, VSAMAIX1, contains the IDCAMS JCL and instructions on how to delete the alternate index path and cluster, define and build the alternate index, and define the path entry.

For more information on STOUTL and the offline utility programs, including the purge utility, see *Connect:Enterprise for z/OS User's Guide. Connect:Enterprise for z/OS Installation Guide* also contains related information about initializing the VSAM batch files during installation when you are creating the VSAM file server.

If the STUTAVIX job detects any errors, you see return codes in report indicating the specific problem. Consult the table below for a list of these return codes.

Return Code	Description/Action to take	Action
0	The alternate index keys maintained in the VCF records have all been verified to be correct. No errors detected.	None
4	The VCF Master Control record indicates the alternate keys have never been populated into the VCF records using the STOUTL PURGE utility. Processing terminates, since it is likely that every VCF record will be reported to have errors.	Execute the PURGE utility. See EXAMPLE member XPURGEX1 for complete instructions on implementing the VCF alternate index; otherwise, do not execute this utility.
8	The "Error Code" column in the report indicates that one or more KEY2 errors were detected. KEY2 represents a second (future use) alternate index key that is also being maintained in the VCF record.	Contact Sterling Commerce Support and forward the report output along with the sequential VCF error output file (//VCFOUT DD) for problem resolution.

Return Code	Description/Action to take	Action
12	The "Error Code" column in the report indicates that one or more KEY1 errors were detected. KEY1 represents the Alternate Index key: [User BatchID + ID + Roll# + Batch#].	Contact Sterling Commerce Support and forward the report output along with the sequential VCF error output file (//VCFOUT DD) for problem resolution. The keys in the VCF must be re-synchronized via the PURGE utility. See EXAMPLE member XPURGEX1 for complete instructions on implementing the VCF alternate index.

Batch Number Maintenance

Batch numbers are automatically assigned to newly added batches (either through online collection or the offline utilities ADD function). Batch numbers are assigned in numerical order. The batch number stays the same for the batch's existence.

After the highest batch number available (specified by the MAXBNO parameter in the PURGE utility) is assigned, the batch numbering process starts over with the first batch number. This is known as batch number rollover. Only unused batch numbers are assigned. Batch numbers assigned to existing batches are not reused until the batch is erased by the ERASE offline utility.

As batch numbers are reused, the rollover count maintained in the VCF master control record increments. This count is also stored in the VPF batch control records as part of the record key. This keeps related batches with the same name in the same order as when they were added, even if a rollover occurred between adds. As a result, batches are selected and transmitted in the same order in which they were received.

Chapter 14

Backing Up Connect:Enterprise

This chapter describes the backup procedure for Connect:Enterprise. This procedure has been designed to provide true round the clock (24 x 7) operation of Connect:Enterprise by allowing the system to be backed up without shutting down the VSAM server.

Understanding the Backup Process

The VSAM repository comprises four files: the VPF file containing file control information and pointers to the batch location in the VBQ file, the VCF file containing the batch control information, the VBQ file containing the actual batches, and the VLF file containing log records. At a minimum the VCF, VPF, and VBQ files must be backed up in synchronization for a backup to be useful.

Connect:Enterprise produces this backup by performing the following actions:

- Stopping all ERASE and MOVE activity using the STUTABKS utility
- Backing up VCF, VPF, VBQ, and VLF files using the backup utility of your choice

Caution: Back up the VCF before the VBQ to protect the integrity of the data.

- Synchronizing the files by recreating the VPF from the VCF using the STUTAPFR utility

Note: Also, see the *Offline Utilities* chapter of *Connect:Enterprise for z/OS for z/OS User's Guide*. This utility validates the VSAM VPF, VCF and VBQ files, and if necessary, resynchronizes them

- Restarting ERASE and MOVE activity using the STUTABKE utility

When ERASE and MOVE activity is stopped, the only changes to the VBQ files are from STOUTL ADD operations and Connect:Enterprise collection processes. The backup of the VCF file reflects the state of the VBQ files at the start of the backup. The existing VPF file is replaced by a new VPF file created from the backup VCF file using the STUTAPFR utility, which synchronizes the files.

Performing a Backup of Connect:Enterprise

To back up Connect:Enterprise without performing a complete system shutdown, perform the following steps:

1. Run STUTABKS. This utility prevents the start of new MOVE and ERASE operations and waits for all in-process MOVE and ERASE activity to stop.

```
//BKUPBEG  EXEC PGM=STUTABKS,PARM=(‘NAME=SRV1,VPF=ENTPRS.VPF,TMR=20’)
```

Note: If a MOVE or ERASE is attempted before these STOUTL utilities have been re-enabled, the functions fail and you receive a CMU209W error message.

The following table contains the responses returned by the STUTABKS utility.

Code	Description
RC=0	STOUTL MOVE and ERASE successfully locked out.
RC=U0600	STOUTL MOVE and ERASE were already locked out or invalid parameters. Message CMUxxxE will show error.

The following table contains the definitions for parameters used by the STUTABKS utility.

Parameter	Description
NAME=xxxx	Required. Specifies the 4 character subsystem name for the Connect:Enterprise VSAM server that is processing the Connect:Enterprise VSAM file you are backing up.
VPF=xxxx.xxxx	Required. Specifies the full data set name of the VPF.
TMR=nnn	Optional. Specifies the time (1–999 seconds) STUTABKS waits between checks to see if currently running STOUTL MOVE or ERASE jobs have completed. The default is 20 seconds.

2. Run the backup utility of your choice to back up the VCF, VPF, VBQ, and VLF files. You ***must*** back up the VCF before the VBQs to ensure the integrity of the files. This requirement allows the STOUTL ADD utility and Connect:Enterprise for z/OS collection processes to run while the backup is in progress. Your backup will reflect the state of the files at the point in time that the VCF is backed up.

3. Run STUTABKE to reenable the MOVE and ERASE utilities.

```
//BKUPEND  EXEC PGM=STUTABKE,PARM=(‘NAME=SRV1,VPF=ENTPRS.VPF’)
```

The following table contains the responses returned by the STUTABKE utility.

Code	Description
RC=0	STOUTL MOVE and ERASE successfully unlocked.
RC=U0600	STOUTL MOVE and ERASE were not locked or invalid parameters. Message CMUxxxE will show error.

The following table contains the definitions for parameters used by STUTABKE

Parameter	Description
NAME=xxxx	Required. Specifies the 4-character subsystem name for the Connect:Enterprise VSAM server that is processing the Connect:Enterprise VSAM file you backed up.
VPF=xxxx.xxxx	Required. Specifies the full data set name of the VPF.

Sample Backup Job

The following sample backup job, found in the VSAMBKUP example member, uses IDCAMS REPRO to back up the VCF, VPF, VLF and VBQ files.

```
//… job card……
//*
//* Run STUTABKS to prevent new STOUTL MOVE or ERASE jobs
//*
//BKUPBEG  EXEC PGM=STUTABKS,PARM=(‘NAME=SRV1,VPF=ENTPRS.VPF,TMR=20’)
//STEPLIB   DD DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//*
//* Run IDCAM REPRO (or backup utility of your choice) to backup VCF, VPF, VLF, VBQ
//*
//BACKUP EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//*
//VPF       DD DSN=ENTPRS.VPF,
//             DISP=(SHR,KEEP,KEEP)
//VPFBKU    DD DSN=ENTRPS.VPF.BKUP,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=140,BLKSIZE=0,DSORG=PS
//*
//VCF       DD DSN=ENTPRS.VCF,
//             DISP=(SHR,KEEP,KEEP)
//VCFBKU    DD DSN=ENTPRS.VCF.BKUP,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=2052,BLKSIZE=0,DSORG=PS
//*
//VLFn      DD DSN=ENTPRS.VLFn,
//             DISP=(SHR,KEEP,KEEP)
//VLFnBKU   DD DSN=ENTPRS.VLFn.BKUP,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=1028,BLKSIZE=0,DSORG=PS
//*
//VBQnn       DD DSN=ENTPRS.VBQnn,
//             DISP=(SHR,KEEP,KEEP)
//VBQnnBKU    DD DSN=ENTPRS.VBQnn.BKUP,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=32756,BLKSIZE=32560,DSORG=PS

//SYSIN     DD *
 REPRO INFILE(VPF) OUTFILE(VPFBKU)
 REPRO INFILE(VCF) OUTFILE(VCFBKU)
 REPRO INFILE(VLFn) OUTFILE(VLFnBKU)
 REPRO INFILE(VBQnn) OUTFILE(VBQnnBKU)
//*
//* Run STUTABKE to allow new STOUTL MOVE or ERASE jobs
//*
//BKUPEND  EXEC PGM=STUTABKE,PARM=(‘NAME=SRV1,VPF=ENTPRS.VPF’)
//STEPLIB   DD DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//*
```

Restoring Connect:Enterprise from a Backup

To restore Connect:Enterprise VSAM server files from a backup that was made using the STUTABKS and STUTABKE programs, perform the following steps:

1. Use the IDCAMS REPRO utility to copy the VPF and VCF to a variable blocked sequential file.
2. Run STUTAPFR to create a current VPF.
3. Sort the new VPF sequential file.
4. Rename the restored VPF file.
5. Define a new VPF with the original name.
6. Use the IDCAMS REPRO utility to copy the variable blocked file into the new VPF file.

Sample STUTAPFR Example Member

The following STUTAPFR example member contains a sample of the steps used to recreate the VPF from a backup of the VCF/VPF.

```
//… job card……
//*  THIS JOB ASSUMES THAT BACKUP VERSIONS OF VPF, VCF, VLF, and VBQ files have
//*  BEEN RESTORED TO ENTPRS.XXX DATASETS PRIOR TO RUNNING THIS JOB.
//*
//CLEANUP EXEC PGM=IEFBR14
//VPFSEQ    DD DSN=ENTPRS.VPF.VBSEQ,
//             DISP=(MOD,DELETE,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(1),RLSE),
//             RECFM=VB,LRECL=140,BLKSIZE=0,DSORG=PS
//VCFSEQ    DD DSN=ENTPRS.VCF.VBSEQ,
//             DISP=(MOD,DELETE,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(1),RLSE),
//             RECFM=VB,LRECL=2052,BLKSIZE=0,DSORG=PS
//VPFOUT    DD DSN=ENTPRS.VPF.VBSEQ.OUT,
//             DISP=(MOD,DELETE,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(1),RLSE),
//             RECFM=VB,LRECL=140,BLKSIZE=0,DSORG=PS
//VPFSCI    DD DSN=ENTPRS.VPF.VBSEQ.SCI,
//             DISP=(MOD,DELETE,DELETE),
//             UNIT=SYSDA,SPACE=(TRK,(1),RLSE),
//             RECFM=VB,LRECL=140,BLKSIZE=0,DSORG=PS
//*
//* IDCAM REPRO RESTORED VPF AND RESTORED VCF FILES TO SEQUENTIAL FILE
//*
//REPROCF EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//*
//VPF       DD DSN=ENTPRS.VPF,
//             DISP=(SHR,KEEP,KEEP)
//VPFSEQ    DD DSN=ENTRPS.VPF.VBSEQ,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=140,BLKSIZE=0,DSORG=PS
//*
//VCF       DD DSN=ENTPRS.VCF,
//             DISP=(SHR,KEEP,KEEP)
//VCFSEQ    DD DSN=ENTPRS.VCF.VBSEQ,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             RECFM=VB,LRECL=2052,BLKSIZE=0,DSORG=PS
//*

//SYSIN     DD *
 REPRO INFILE(VPF) OUTFILE(VPFSEQ)
 REPRO INFILE(VCF) OUTFILE(VCFSEQ)
```

Continued

```
//*
//* RECREATE VPF RECORDS FROM RESTORED VCF/VPF
//*
//BLDVPF  EXEC PGM=STUTAPFR
//STEPLIB   DD DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//VCF       DD DISP=SHR,DSN=ENTRPS.VCF.VBSEQ
//VPFIN     DD DISP=SHR,DSN=ENTPRS.VPF.VBSEQ
//VPFOUT    DD DSN=ENTPRS.VPF.VBSEQ.OUT,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//             DSORG=PS,RECFM=VB,LRECL=140,BLKSIZE=0
//*
//*
//* SORT GENERATED VPF RECORDS
//*
//* NOTE: CHANGE SYNCSORT TO THE PROPER SORT PROGRAM FOR YOUR
//* ENVIRONMENT.
//*
//SORTVPF EXEC PGM=SYNCSORT
//SORTWRK1 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWRK2 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTWRK3 DD UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE)
//SORTIN   DD DISP=SHR,DSN=ENTPRS.VPF.VBSEQ.OUT
//SORTOUT  DD DSN=ENTPRS.VPF.VBSEQ.SCI,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(CYL,(PP,SS),RLSE),
//            DCB=ENTPRS.VPF.VBSEQ.OUT
//SYSIN    DD *
 SORT FIELDS=(1,17,BI,A)
//SYSOUT   DD   SYSOUT=*
//SYSPRINT DD DUMMY
//****************************************************************
//*  RENAME ORIGINAL VPF FILE                                    *
//****************************************************************
//RENAME  EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
 ALTER     'ENTPRS.VPF.INDEX'             +
   NEWNAME('ENTPRS.VPF.INDEX.OLD')
 ALTER     'ENTPRS.VPF.DATA'              +
   NEWNAME('ENTPRS.VPF.DATA.OLD')
 ALTER     'ENTPRS.VPF'                   +
   NEWNAME('ENTPRS.VPF.OLD')
//*
```

Continued

```
//*
//****************************************************************
//*  ALLOCATE NEW VPF FILE USING OLD FILE NAME                    *
//*                                                              *
//*  CHANGE pppppp and ssssss to the same values used when       *
//*  defining the original VPF during installation.              *
//****************************************************************
//ALLOC   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//XXXXXXXX   DD    UNIT=SYSDA,VOL=SER=XXXXXXXX,DISP=SHR
//SYSIN     DD *
/*******   DEFINE VPF FILE    ********/
 DEFINE CLUSTER -
   (NAME(ENTPRS.VPF) -
   RECORDS (PPPPPP SSSSSS)  /* MINIMUM (MAXBNO VALUE X 10) + 25 */ -
   VOLUMES(XXXXXX) -
   REUSE -
   SHAREOPTIONS(2) -
   KEYS(17 0) -
   RECSZ(136 136)) -
 DATA -
   (NAME(ENTPRS.VPF.DATA)) -
 INDEX -
   (NAME(ENTPRS.VPF.INDEX))
//*
…
//*
//****************************************************************
//*  REPRO GENERATED VPF FILE INTO NEW VPF FILE STRUCTURE        *
//****************************************************************
//REPRO   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//*
//VPFSCI    DD DSN=ENTPRS.VPF.VBSEQ.SCI,
//             DISP=(SHR,KEEP,KEEP)
//*
//VPF       DD DSN=ENTPRS.VPF,
//             DISP=(SHR,KEEP,KEEP)
//*
//SYSIN     DD *
 REPRO INFILE(VPFSCI) OUTFILE(VPF)
//*
```

Chapter 15

Browsing Data

A browse data space is an address space that holds viewable user data, not control blocks or executable programs. This data space can be browsed using the CICS and ISPF interfaces, and is controlled by parameters set in the options definition file (ODF). Each space can be allocated with from 1 to 524288 4K pages (4KB–2GB), and once allocated, the size of the data space cannot be increased. Approximately 480 data spaces can be created and placed in the PASN access list accessible by any task in the address space. The PASN access list is used because it allows multiple CP tasks to service browse requests. PASN data spaces are automatically deleted when the address space that owns them goes away. For example, if an instance of Connect:Enterprise is shut down, all of its assigned data spaces are automatically deleted, and the resources used by them are released.

Configuring the Browse Data Space Feature

Separating the VSAM I/O and the SNA I/O is a key component of the browse data space facility. A data space only needs to be loaded once, and as long as it is not deleted, multiple users can browse it multiple times without incurring any more VSAM I/O. Each browse request establishes a separate session with the data space.

Browsing a batch using this facility is a two-phase process. First the batch is read from the VBQ and written to the data space. If the batch has certain attributes (offline or FTP added with STRUCTURE=FILE and non-transparent) the data is broken into logical records as it is being written to the data space. Second, the batch is read from the data space, formatted into STIPS, and transmitted to the requesting online interface using SNA LU 6.2.

Five ODF parameters, as shown in the following table, control the browse data space facility.

Parameter	Valid Values	Default	Units
BROWSE_AUTOCLEAN_INTERVAL	0–32767	60	Seconds
BROWSE_DATASPACE_COUNT_MAX	0–480	20	Data spaces

Parameter	Valid Values	Default	Units
BROWSE_DATASPACE_SIZE_MAX	1–524288	524288	4K pages
BROWSE_SESSION_COUNT_MAX	1–1023	40	Sessions
BROWSE_SESSION_RETIREMENT_AGE	0–32767	300	Seconds

Browse Operating Modes

There are four distinct operating modes for using the browse data space facility. They are controlled by defining the ODF parameters as described in the following paragraphs.

Legacy Mode

Set the BROWSE_DATASPACE_COUNT_MAX parameter to 0 (zero).

This mode disables the entire function causing the browse request to conform to prior versions of Connect:Enterprise. All other BROWSE parameters still go through ODF validation, but they are otherwise ignored and no data spaces are created. The new CICS 22.1 line command set to 0 (zero) invokes the browse interface without a data space, displaying entire VBQ blocks.

Auto Clean Timer Mode

None of the ODF browse parameters are set to 0 (zero). This is the default mode.

Auto clean means a session is deleted if it is inactive for the number of seconds specified in the BROWSE_SESSION_RETIREMENT_AGE parameter. If a deleted session is the last one using a particular data space, the data space is also deleted. If a session or data space is deleted for this reason, it is said to be retired. In auto clean timer mode the automatic cleanup is initiated in two ways.

- At the end of each call to the browse data space API module, the API attempts to perform the auto clean function. If after that, one or more data spaces still exist, the auto clean timer is set to repeat the attempt in the number of seconds specified in the BROWSE_AUTOCLEAN_INTERVAL. If all data spaces are retired, the timer is not set.
- The auto clean timer attempts to perform the auto clean function. If after that attempt, one or more data spaces still exist, the auto clean timer is set to repeat the attempt in the number of seconds specified in the BROWSE_AUTOCLEAN_INTERVAL. If all data spaces are retired, the timer is not set.

Note: Each CP task can have its own timer, and thus can initiate an attempt to auto clean.

Timerless Auto Clean Mode

Set the BROWSE_AUTOCLEAN_INTERVAL parameter to 0 (zero).

The auto clean function occurs only at the end of each call to the browse data space API module. If there are one or more data spaces in existence, cleanup will wait for the next call to the API. The next call can be for the same session, or a different session.

Auto Clean Disabled Mode

Set the BROWSE_AUTOCLEAN_INTERVAL parameter to 0 (zero), and the BROWSE_SESSION_RETIREMENT_AGE parameter to 0 (zero).

Using these settings, auto clean never occurs. The number of browse data spaces increases until it reaches the value set in the BROWSE_DATASPACE_COUNT_MAX parameter. Thereafter, when a new data space is requested, the oldest unused browse data space is deleted, along with all its sessions. A session or data space deleted for this reason is said to be stolen.

Though the overhead of deleting and creating a data space is insignificant, the cost of loading it with data can be high. If stealing occurs often, increase the value set in the BROWSE_DATASPACE_COUNT_MAX parameter.

Performance Tuning with the Browse Data Space Parameters

The single most important factor in browse performance in any operating mode is batch size. The LU 6.2 transmission of data from the Connect:Enterprise main task to the data space can create a bottleneck. The CICS 22.1 line command set to 0 (zero) only retrieves enough data to display one screen at a time, not the entire batch. (The CICS 22.1 mode is much faster than any other.) When using the line command set to 1 (one), the only way to reduce the bottleneck is to reduce the size of the batch.

The next most important factor in tuning browse performance, is the number of VBQ blocks needed to contain the data. If the batch is VBQ blocked, the block count is an insignificant factor in comparison with the number of bytes. But if the batch is VBQ unblocked, the VSAM I/O can create a bottleneck. The way to reduce this type of bottleneck is to use VBQ blocking (the default setting).

The auxiliary storage manager can create a performance bottleneck, usually as a result of a large batch swamping the local page data sets. The maximum number of allocatable 2 GB data spaces is 480, but few systems have the capability to support the maximum number at one time. Use the BROWSE_DATASPACE_SIZE_MAX parameter to limit the number of data spaces that can exist at one time. Use the BROWSE_DATASPACE_SIZE_MAX parameter to limit the size of each data space.

Finally, the least important tuning consideration is the number of concurrent sessions allowed. The BROWSE_SESSION_COUNT_MAX parameter cannot be less than the value set in the BROWSE_DATASPACE_COUNT_MAX, but it can safely be set to its maximum value (1023). If it is set too low, sessions can be stolen.

Chapter 16

Diagnostics

This chapter describes how to diagnose problems with Connect:Enterprise. It contains the following topics:

- Diagnosing problems
- Online traces
- FTP session dialog trace

Diagnosing Problems

Perform the following actions to diagnose and resolve problems with Connect:Enterprise.

- Verify that your telecommunications equipment and lines are functioning properly. Test an alternate equipment configuration and use different communications lines to isolate the problem. Use a Connect:Enterprise trace of communications activity to verify your hardware. Traces are listed in *Online Traces* on page 372.
- Check the system console for messages. Look up the meaning and resolution of any message in the *Connect:Enterprise for z/OS Messages and Codes Guide*.
- Examine the Connect:Enterprise SNAPOUT and BTSNAP data sets.

  Your execution JCL should always contain a SNAPOUT DD statement and a BTSNAP DD statement. These files contain snapshot dumps of certain Connect:Enterprise control blocks when severe errors occur. Each snapshot dump contains a descriptive title to explain the problem. These dumps are particularly helpful to a Sterling Commerce Customer Support representative.

  New Connect:Enterprise customers should always print and examine the SNAPOUT data sets when first installing and using online Connect:Enterprise. Discuss any unusual messages with a Sterling Commerce Customer Support representative.
- Verify the ODF values. For more information, see the *Connect:Enterprise for z/OS Administration Guide*.
- Verify the user assembly (BSC Only). For more information, see the *Connect:Enterprise for z/OS Installation Guide.*

- Check your STEPLIB library usage. Verify that the correct versions of all Connect:Enterprise load modules are present. Verify that the VSAM file server executes from an APF-authorized library.
- Examine the SYSOUT files from the FTP Dialog trace for FTP problems.
- Examine SYSPRINT if you ran any TCP scheduler traces.

If none of these actions resolve the problem, contact Sterling Commerce Customer Support.

Online Traces

Online traces can help you analyze and resolve Connect:Enterprise problems. Contact a Sterling Commerce Customer Support representative to help interpret the trace output, if necessary.

Caution: Do not run a trace on a production Connect:Enterprise system unless a Sterling Commerce Customer Support representative asks you to.

Do one of the following to activate a trace:

- Add a record to the *OPTIONS records in the ODF before submitting online Connect:Enterprise for execution.

  Tracing begins immediately after Connect:Enterprise is brought up. After you recreate and trace the problem, shut down Connect:Enterprise and remove the trace record from the ODF.
- Enter a $$TRACE command from the system console. The $$TRACE command overrides any traces defined in the ODF.
- Submit a trace from a CICS or ISPF interface panel.

The following table describes the different traces and the console commands and ODF parameters to run them. (See the *Connect:Enterprise for z/OS ISPF User’s Guide* and the *Connect:Enterprise for z/OS CICS User’s Guide* to run traces from those interfaces.)

Information Traced	ODF Parameter	Console Command	Description
All SNA or BSC I/O completions	TRACE=ALLTP	$$TRACE ON|OFF	Traces all active SNA sessions or BSC lines I/O completions. To interpret trace information, you must be familiar with VTAM RPL/BTAM DECB format. Perform the following to find the VTAM RPL/BTAM DECB in the trace: ◆ The first 16 bytes of trace data contain a control block identifier followed by the Connect:Enterprise version/release/ maintenance number. The next 8 bytes contain the SNA remote name or BSC line ID where an I/O completion occurred. The next 11 full words contain internal status bytes. These are followed by a full word containing the address of the VTAM RPL or the BTAM DECB. This address points further down the trace output to the actual RPL or DECB. ◆ If you can read an Assembly Language DSECT, the M$LCB macro in the Connect:Enterprise source library can help you interpret the trace. The trace output contains the Line Control Block (LCB). ◆ If you need help in finding or interpreting this information, contact Sterling Commerce Customer Support.
SNA activity	TRACE=SNA	$$TRACE SNAON\| SNAOFF	Traces all SNA logons and unusual SNA activity, such as invalid FMHs, session outages, deblocking errors, and logon rejections. Use this option if you install and test the SNA component of a new Connect:Enterprise system. This trace produces a much smaller output data set than using just the $$TRACE command or TRACE=ALLTP ODF parameter.
Connect:Enterprise line or region status	N/A	$$DUMP LINEID= xxxxxxx|ALL	Provides a DUMP of the current Connect:Enterprise BSC line status or the entire Connect:Enterprise region status. Replace xxxxxxxx with the line ID (defined in the M$LINXEX section in the User Assembly) or remote name to trace.

Information Traced	ODF Parameter	Console Command	Description
VSAM functions	TRACE=VSAM	$$TRACE VSAMON| VSAMOFF	Traces all call requests to the Connect:Enterprise VSAM batch files handler. This provides a general picture of all VSAM functions. Run this trace if requested by a Sterling Commerce Customer Support representative, and forward the trace output to Sterling Commerce.
VSAM file server	TRACE=VA2C	$$TRACE VA2CON| VA2COFF	Traces all call requests to the Connect:Enterprise VSAM file server. The request parameters and the completion code are formatted. Run this trace if requested by a Sterling Commerce Customer Support representative, and forward the trace output to Sterling Commerce.
BTKMGR functions	TRACE=A2C	$$TRACE A2CON| A2COFF	Traces all requests made to the BTKMGR service routines. The request parameters and the completion code are formatted. Run this trace if requested by a Sterling Commerce Customer Support representative, and forward the trace output to Sterling Commerce.
I/O completions for a single SNA or BSC line	TRACEID= xxxxxxxx	N/A	Traces I/O completions for a single SNA remote name or BSC line ID. Replace xxxxxxxx with the line ID or remote name to trace. Use this parameter instead of the TRACE=ALLTP ODF parameter to limit the amount of trace output. It does not have a console command equivalent.
User exits	TRACE=EXITS	$$TRACE EXITON| EXITOFF	Traces information passed to and from user-supplied exit programs. This trace is only valid for online Connect:Enterprise user exits. Trace parameters are passed to and from the user exit programs, before and after each call to the exit. The trace output contains the Exit Control Block (XCB).
APPC	TRACE=APO	$$TRACE APOON| APOOFF	Traces all APPC LU 6.2 macro completions.
APPC	TRACE=APQ	$$TRACE APQON| APQOFF	Traces information passed between the APPC LU 6.2 task and the process router task.

Information Traced	ODF Parameter	Console Command	Description
CP tasks	TRACE=CP	$$TRACE CPON| CPOFF	Traces information passed to and from CP tasks. The trace output helps diagnose APPC activity from any APPC remote, including the ISPF and CICS interfaces.
FTP activity	TRACE_FTP=*| remotename [,list]	$$TRACE FTPON=*\| remotename[,list] [,c] $$TRACE FTPOFF=*| remotename[,list] [,c]	**ODF parameter:** The TRACE_FTP=* ODF record activates tracing on all FTP sessions. The TRACE_FTP=|remotename[,list] ODF record activates tracing for only specified individual remote names. You can combine the wildcard and specific remote names. For example: TRACE_FTP=RMT1*,RMT234,RMT88 traces FTP activity on all remote sites beginning with RMT1 and on specific remote sites RMT234 and RMT88. You can specify multiple TRACE_FTP ODF records to trace multiple FTP sessions. For example: TRACE_FTP=RMT001,RMT011,RMT111 TRACE_FTP=RMT002,RMT022,RMT222 TRACE_FTP=RMT003,RMT033,RMT333 **Console command:** The $$TRACE FTPON=* console command activates tracing on all TP sessions. The $$TRACE FTPON=remotename[,list] command activates tracing for only the specified individual remote names. The c option of the $$TRACE command traces SSL data as unencrypted. The default is to trace SSL data as encrypted. The $$TRACE FTPOFF command turns off session dialog tracing. **Note:** All FTP remote trace activity is written to the DIALOG file (DTnnnnnn). In addition to these traces, you can also capture FTP dialog information. See *FTP Session Dialog Trace* on page 377 for more information.
TCP scheduler activity	TRACE=TCPSCH	$$TRACE TCPSCHON| TCPSCHOFF	Traces TCP scheduler activity. **Note:** TSPSCH trace activity is written to SYSPRINT.

Information Traced	ODF Parameter	Console Command	Description
TCP scheduler activity	TCPSCH=xxxxxxx	$$TRACE TCPSCHRMT= xxxxxxx	This value traces TCP scheduler activity for a single remote. **Note:** TRACE=TCPSCH must be active to use this feature.
Console application agents	TRACE=RPCON	$$TRACE RPCON| RPCOFF	This value traces all activity processing for all console application agent requests. For more information on application agents, see the *Connect:Enterprise for z/OS Application Agents and User Exits Guide*. You can start separate traces for each application agent.
Scheduler application agents	TRACE=RPSON	$$TRACE RPSON| RPSOFF	This value traces all activity processing for all scheduler application agent requests.
End-of-Batch application agents	TRACE=RPEOB	$$TRACE RPEON| RPEOFF	This value traces all activity processing for all end-of-batch application agent requests.
Logging application agents	TRACE=RPLOG	$$TRACE RPLON| RPLOFF	This value traces all activity processing for all logging application agent requests.
Wake up terminate application agents	TRACE=RPWKT	$$TRACE RPWON| RPWOFF	This value traces all activity processing for all wake up terminate application agent requests.
Process router transactions	TRACE=PR	$$TRACE PRON| PROFF	Traces information passed to and from the process router. The process router is a program that routes transactions to and from the CICS and ISPF interfaces. It also routes application agent rules requests for processing. This trace can help diagnose APPC transaction problems.

Trace Output

Trace output is written to a SNAPOUT data set. Use the following JCL to print the SNAPOUT data set:

```
//PRTSNAPJOB  ...  AS REQUIRED FOR YOUR SITE
//***********************************************************
//*    PRINT Connect:Enterprise SNAPSHOT DATA SET
//***********************************************************
//*
//PRTSNAP   EXEC   PGM=IEBGENER
//SYSPRINT  DD     SYSOUT=*
//SYSUT1    DD     DSN=ENTPRS.SNAPOUT,DISP=SHR
//SYSUT2    DD     SYSOUT=*
//SYSIN     DD     DUMMY
```

If you write the SNAPOUT data set directly to SYSOUT, ensure that the online Connect:Enterprise execution JCL specifies the following DCB information:

```
//SNAPOUT DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=1632)
```

Allocate application agent traces to the RULTRACE data set. See the *Tracing Application Agent Requests* section of the *Connect:Enterprise for z/OS Application Agents and User Exits Guide* for more information.

FTP Trace Output

All FTP remote trace activity is written to the DIALOG file (DTnnnnnn). The following is an example of the output from an FTP remote trace:

```
09:49:06:60  FTP DATA RECEIVED:
09F60000  6161E2E5 C1D1C4F2 C1C440D1 D6C2404D  C3D4D4C1 C9D5C45D 6B7DE240 E5C1D1C4  *//SVAJD2AD JOB (CMMAIND),'S VAJD*
09F60020  C17D6BD4 E2C7C3D3 C1E2E27E E76BC3D3  C1E2E27E E76BD5D6 E3C9C6E8 7EE2E5C1  *A',MSGCLASS=X,CLASS=X,NOTIFY=SVA*
09F60040  D1C4F215 615CD1D6 C2D7C1D9 D44040E2  E8E2C1C6 C67EC3E2 C7C11561 61E4E3C9  *JD2./*JOBPARM  SYSAFF=CSGA.//UTI*
09F60060  D3F14040 4040C5E7 C5C340D7 C7D47EE2  E3D6E4E3 D36BD7C1 D9D47E7D E2D1E5C1  *L1    EXEC PGM=STOUTL,PARM='SJVA*
09F60080  7D6BD9C5 C7C9D6D5 7EF2D415 61615CE3  C5D7D3C9 C24040C4 C4404040 C4C9E2D7  *',REGION=2M.//*TEPLIB  DD   DISP
09F600A0  7EE2C8D9 6BC4E2D5 7ED4C2E7 C4C5E54B  E5F4F0F0 E2D1E54B D3D6C1C4 156161E2  *=SHR,DSN=MBXDEV.V400SJV.LOAD.//S*
09F600C0  E3C5D7D3 C9C24040 C4C44040 40C4C9E2  D77EE2C8 D96BC4E2 D57ED4C2 E7C4C5E5  *TEPLIB  DD   DISP=SHR,DSN=MBXDEV*
09F600E0  4BE5F4F0 F0E3C5E2 E34BD3D6 C1C41561  61C9D5E3 D9C4D940 4040C4C4 404040E2  *.V400TEST.LOAD.//INTRDR   DD   S*
09F60100  E8E2D6E4 E37E4DC1 6BC9D5E3 D9C4D95D  156161E2 E8E2E3C5 D9D44040 C4C44040  *YSOUT=(A,INTRDR).//SYSTERM  DD  *
09F60120  40E2E8E2 D6E4E37E 5C156161 C2E3E2D5  C1D74040 40C4C440 4040E2E8 E2D6E4E3  * SYSOUT=*.//BTSNAP   DD   SYSOUT*
09F60140  7E5C6BC4 C3C27E4D D9C5C3C6 D47EC6C1  6BD3D9C5 C3D37EF1 F3F36BC2 E4C6D5D6  *=*,DCB=(RECFM=FA,LRECL=133,BUFNO
09F60160  7EF05D15 6161D7D9 C9D5E340 404040C4  C4404040 E2E8E2D6 E4E37E5C 6BC4C3C2  *=0).//PRINT    DD   SYSOUT=*,DCB*
09F60180  7E4DD9C5 C3C6D47E C6C16BD3 D9C5C3D3  7EF1F3F3 6BC2E4C6 D5D67EF0 5D156161  *=(RECFM=FA,LRECL=133,BUFNO=0).//*
09F601A0  E2E8E2D7 D9C9D5E3 40C4C440 4040E2E8  E2D6E4E3 7E5C6BC4 C3C27E4D D9C5C3C6  *SYSPRINT DD   SYSOUT=*,DCB=(RECF*
09F601C0  D47EC6C1 6BD3D9C5 C3D37EF1 F3F36BC2  E4C6D5D6 7EF05D15 6161D9C5 D7D6D9E3  *M=FA,LRECL=133,BUFNO=0).//REPORT*
09F601E0  E24040C4 C4404040 E2E8E2D6 E4E37E5C  6BC4C3C2 7E4DD9C5 C3C6D47E C6C16BD3  *S  DD   SYSOUT=*,DCB=(RECFM=FA,L*
09F60200  D9C5C3D3 7EF1F3F3 6BC2E4C6 D5D67EF0  5D156161 5C156161 E2E8E2C9 D5404040  *RECL=133,BUFNO=0).//*.//SYSIN   *
09F60220  40C4C440 4040C4C1 E3C16BC4 D3D47E7C  7C155C15 404040C5 D9C1E2C5 15404040  * DD   DATA,DLM=@@.*.   ERASE.   *
09F60240  404040E5 D7C67E7D C3E2C4D4 C2E74BE2  D1E5F4F0 F0E34BE5 D7C67D15 5C404040  *   VPF='CSDMBX.SJV400T.VPF'.*   *
09F60260  4040C9C4 7EE3C5E2 E3D3C9E2 E3154040  40404040 E5C2D87E F4155C40 40404040  *  ID=TESTLIST.      VBQ=4.*     *
09F60280  C2C1E3C3 C8D5E4D4 7EF160F5 F0155C40  40404040 C2C1E3C3 C8C9C47E 7DF2F040  *BATCHNUM=1-50.*     BATCHID='20 *
09F602A0  D9C5C3D6 D9C4E240 6040E5C2 D8C2D3D6  C3D2C5C4 7D157C7C 15                *RECORDS - VBQBLOCKED'.@@.       *
```

Note: The FTP remote online trace can be combined with the FTP session dialog trace to produce a trace of all FTP dialog and data. See *Dialog Trace Output* on page 378 for more information.

FTP Session Dialog Trace

Connect:Enterprise FTP can trace the conversation, or dialog, between the FTP client and the FTP Server. The dialog trace differs from a usual Connect:Enterprise trace in that a dialog trace presents dialog information; it does not capture or present program execution information.

Like a trace, you specify the FTP session dialog trace by an *OPTIONS parameter, a console command, or through an ISPF or CICS interface panel. The following table describes FTP session dialog console commands and ODF parameters. (See the *Connect:Enterprise for z/OS ISPF User's Guide* and *Connect:Enterprise for z/OS CICS User's Guide* to run the FTP session dialog trace from those interfaces.)

Information Traced	ODF Parameter	Console Command	Description
FTP session dialog	DIALOG__FTP=*|remotename[,list]	$$DIALOG FTPON=*\|remotename[,list] $$DIALOG FTPOFF=*|remotename[,list]	**ODF parameter:** The DIALOG__FTP=* ODF record activates FTP session dialog tracing for all FTP sessions. The DIALOG__FTP=|remotename[,list] ODF record activates FTP session dialog tracing for only specified individual remote names. You can combine the wildcard and specific remote names. For example: DIALOG__FTP=RMT1*,RMT234,RMT88 activates FTP session dialog tracing on all remote sites beginning with RMT1 and on specific remote sites RMT234 and RMT88. You can specify multiple DIALOG__FTP ODF records to trace multiple FTP session dialogs. For example: DIALOG__FTP=RMT001,RMT011,RMT111 DIALOG__FTP=RMT002,RMT022,RMT222 DIALOG__FTP=RMT003,RMT033,RMT333 **Console command:** The $$DIALOG FTPON=* command activates FTP session dialog tracing for all FTP sessions. The $$DIALOG FTPON=remotename[,list] command activates FTP session dialog tracing for only specified individual remote names. The $$DIALOG FTPOFF command turns off session dialog tracing.

Dialog Trace Output

Connect:Enterprise refers to each FTP session as a process and assigns it a unique 6-digit process number ranging from 000001 to 999999. (The process number resets to 000001 when it exceeds 999999).

When you activate a session dialog trace, Connect:Enterprise dynamically allocates a unique SYSOUT file for each FTP session. For example, if there are five FTP sessions, five unique SYSOUT files are allocated. The DDNAME assigned to each allocated SYSOUT file consists of the characters “DT” concatenated with a unique dialog trace number. This number is incremented by one for each SYSOUT that is opened.

Note: The REXX variable, RDXVARS, which contains a list of all Connect:Enterprise REXX variables except itself, the REPLY., DIR., and LOCDIR. variables, includes a variable called DTDDNAME for the FTP client. This variable contains the ddname of the FTP client session’s dialog trace DD if dilaog trace is active for the remote. Otherwise, this variable is blank.

Each session’s dialog information is written to its SYSOUT file as individual records. If the record data cannot fit on one line, the data continues on as many records as necessary.

The following is a sample dialog trace:

```
08:40:37:90  FTP CLIENT INPUT:  USER FTPRMT01.
08:40:37:90  FTP SERVER OUTPUT: 331 Send password please.
08:40:37:91  FTP CLIENT INPUT:  PASS TESTPW.
08:40:37:94  FTP SERVER OUTPUT: 230 FTPRMT01 is logged on.  Current working Mailbox is "FTPRMT01".
08:40:37:95  FTP CLIENT INPUT:  PASS ********
08:40:37:95  FTP SERVER OUTPUT: 200 Data representation type is E.
08:40:38:40  FTP CLIENT INPUT:  SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=23440.
08:40:38:40  FTP SERVER OUTPUT: 200 SITE command was accepted.
08:40:38:41  FTP CLIENT INPUT:  PORT 199,1,4,2,4,109.
08:40:38:41  FTP SERVER OUTPUT: 200 PORT request OK (199,1,4,2,4,109).
08:40:38:42  FTP CLIENT INPUT:  STOR FTPRMT01.
08:40:38:51  FTP SERVER OUTPUT: 150 Opening data connection.  Storing 'FTPRMT01.#0000353'.
08:40:41:20  FTP SERVER OUTPUT: 226 Transfer complete.  'FTPRMT01.#0000353'      697 bytes.
08:40:41:21  FTP CLIENT INPUT:  QUIT.
08:40:41:21  FTP SERVER OUTPUT: 221 QUIT command received. Goodbye.
08:40:41:23  CLOSING TRACE
```

The following table shows the data record layout contents:

Starting Column	Ending Column	Length	Description
008	018	11	The time stamp in hours:minutes:seconds:hundreths of seconds.
021	033	18	Either FTP CLIENT INPUT (the inbound command) or FTP SERVER OUTPUT (the outbound command).
040	Variable	Variable	The data received from or sent to the FTP remote.

You can specify both FTP session dialog trace and the FTP remote trace (the $$TRACE FTPON console command or TRACE_FTP ODF parameter) to produce a complete trace of all FTP dialog and data. This combined trace is written to a SYSOUT file (DTnnnnnn). The following is a sample FTP trace combining the FTP session dialog trace and an online FTP trace:

```
08:31:23:92  FTP CLIENT INPUT:  USER FTPRMT01.
08:31:23:92  FTP SERVER OUTPUT: 331 Send password please.
08:31:23:93  FTP CLIENT INPUT:  PASS TESTPW.
08:31:23:99  FTP SERVER OUTPUT: 230 FTPRMT01 is logged on.  Current working Mailbox is "FTPRMT01".
08:31:24:00  FTP CLIENT INPUT:  PASS ********
08:31:24:00  FTP SERVER OUTPUT: 200 Data representation type is E.
08:31:24:46  FTP CLIENT INPUT:  SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=23440.
08:31:24:46  FTP SERVER OUTPUT: 200 SITE command was accepted.
08:31:24:48  FTP CLIENT INPUT:  PORT 199,1,4,2,4,105.
08:31:24:48  FTP SERVER OUTPUT: 200 PORT request OK (199,1,4,2,4,105).
08:31:24:50  FTP CLIENT INPUT:  STOR FTPRMT01.
08:31:24:60  FTP SERVER OUTPUT: 150 Opening data connection.  Storing 'FTPRMT01.#0000352'.
08:31:24:61  FTP DATA RECEIVED:
09F60000  6161E2E5 C1D1C4F2 C1C440D1 D6C2404D   C3D4D4C1 C9D5C45D 6B7DE240 E5C1D1C4  *//SVAJD2AD JOB (CMMAIND),'S VAJD*
09F60020  C17D6BD4 E2C7C3D3 C1E2E27E E76BC3D3   C1E2E27E E76BD5D6 E3C9C6E8 7EE2E5C1  *A',MSGCLASS=X,CLASS=X,NOTIFY=SVA*
09F60040  D1C4F215 615CD1D6 C2D7C1D9 D44040E2   E8E2C1C6 C67EC3E2 C7C11561 61E4E3C9  *JD2./*JOBPARM  SYSAFF=CSGA.//UTI*
09F60060  D3F14040 4040C5E7 C5C340D7 C7D47EE2   E3D6E4E3 D36BD7C1 D9D47E7D E2D1E5C1  *L1    EXEC PGM=STOUTL,PARM='SJVA*
09F60080  7D6BD9C5 C7C9D6D5 7EF2D415 61615CE3   C5D7D3C9 C24040C4 C4404040 C4C9E2D7  *',REGION=2M.//*TEPLIB  DD   DISP*
09F600A0  7EE2C8D9 6BC4E2D5 7ED4C2E7 C4C5E54B   E5F4F0F0 E2D1E54B D3D6C1C4 156161E2  *=SHR,DSN=MBXDEV.V400SJV.LOAD.//S*
09F600C0  E3C5D7D3 C9C24040 C4C44040 40C4C9E2   D77EE2C8 D96BC4E2 D57ED4C2 E7C4C5E5  *TEPLIB  DD   DISP=SHR,DSN=MBXDEV*
09F600E0  4BE5F4F0 F0E3C5E2 E34BD3D6 C1C41561   61C9D5E3 D9C4D940 4040C4C4 404040E2  *.V400TEST.LOAD.//INTRDR   DD   S*
09F60100  E8E2D6E4 E37E4DC1 6BC9D5E3 D9C4D95D   156161E2 E8E2E3C5 D9D44040 C4C44040  *YSOUT=(A,INTRDR).//SYSTERM  DD  *
09F60120  40E2E8E2 D6E4E37E 5C156161 C2E3E2D5   C1D74040 40C4C440 4040E2E8 E2D6E4E3  * SYSOUT=*.//BTSNAP   DD   SYSOUT*
09F60140  7E5C6BC4 C3C27E4D D9C5C3C6 D47EC6C1   6BD3D9C5 C3D37EF1 F3F36BC2 E4C6D5D6  *=*,DCB=(RECFM=FA,LRECL=133,BUFNO*
09F60160  7EF05D15 6161D7D9 C9D5E340 404040C4   C4404040 E2E8E2D6 E4E37E5C 6BC4C3C2  *=0).//PRINT    DD   SYSOUT=*,DCB*
09F60180  7E4DD9C5 C3C6D47E C6C16BD3 D9C5C3D3   7EF1F3F3 6BC2E4C6 D5D67EF0 5D156161  *=(RECFM=FA,LRECL=133,BUFNO=0).//*
09F601A0  E2E8E2D7 D9C9D5E3 40C4C440 4040E2E8   E2D6E4E3 7E5C6BC4 C3C27E4D D9C5C3C6  *SYSPRINT DD   SYSOUT=*,DCB=(RECF*
09F601C0  D47EC6C1 6BD3D9C5 C3D37EF1 F3F36BC2   E4C6D5D6 7EF05D15 6161D9C5 D7D6D9E3  *M=FA,LRECL=133,BUFNO=0).//REPORT*
09F601E0  E24040C4 C4404040 E2E8E2D6 E4E37E5C   6BC4C3C2 7E4DD9C5 C3C6D47E C6C16BD3  *S  DD   SYSOUT=*,DCB=(RECFM=FA,L*
09F60200  D9C5C3D3 7EF1F3F3 6BC2E4C6 D5D67EF0   5D156161 5C156161 E2E8E2C9 D5404040  *RECL=133,BUFNO=0).//*.//SYSIN   *
09F60220  40C4C440 4040C4C1 E3C16BC4 D3D47E7C   7C155C15 404040C5 D9C1E2C5 15404040  * DD   DATA,DLM=@@.*.   ERASE.   *
09F60240  404040E5 D7C67E7D C3E2C4D4 C2E74BE2   D1E5F4F0 F0E34BE5 D7C67D15 5C404040  *   VPF='CSDMBX.SJV400T.VPF'.*   *
09F60260  4040C9C4 7EE3C5E2 E3D3C9E2 E3154040   40404040 E5C2D87E F4155C40 40404040  *  ID=TESTLIST.      VBQ=4.*     *
09F60280  C2C1E3C3 C8D5E4D4 7EF160F5 F0155C40   40404040 C2C1E3C3 C8C9C47E 7DF2F040  *BATCHNUM=1-50.*     BATCHID='20 *
09F602A0  D9C5C3D6 D9C4E240 6040E5C2 D8C2D3D6   C3D2C5C4 7D157C7C 15                 *RECORDS - VBQBLOCKED'.@@.       *
08:31:25:49  FTP SERVER OUTPUT: 226 Transfer complete.  'FTPRMT01.#0000352'         697 bytes.
08:31:25:50  FTP CLIENT INPUT:  QUIT.
08:31:25:50  FTP SERVER OUTPUT: 221 QUIT command received. Goodbye.
08:31:25:53  CLOSING TRACE
```

Reason Codes and Messages Issued When Scanning is Disabled

The following table describes the messages and codes written to the dialog trace when scanning is disabled during RETR or STOR/STOU processing by means other than the SITE or LOCSITE command. None of these messages require action, but they are included in dialog traces so that you can see how the $$ADD commands have been processed.

Reason Code	Dialog Trace Message Text	Description
1	No command detected and Mode/Stru/Type=S/F/I	For a file data structure transferred in stream mode containing images (STRU F, MODE S,TYPE I or “SFI”), a command was detected but it had no record delimiter, that is, another command or $$END. Once the data in the file begins at the point where no $$ command is detected, there is no way to detect another $$ command, since there is no record delimiter that can be recognized. Any $$ commands in an SFI transmission must be first in the data and delimited by another command or $$END. After the first $$ADD command is recognized, scanning is turned off and the rest of the file is treated as data, including any subsequent $$ADD commands.

Reason Code	Dialog Trace Message Text	Description
2	No command detected in the current record and $$ADD was not detected in prior records	Once the data in the file begins at the point where no $$ command is detected, scanning is turned off if there was no prior $$ADD. Scanning works the same way in SNA and BSC.
3	Command with no delimiter was detected and batch has File Structure. Command treated as data.	The data is being transmitted using MODE=S, STRU=FILE and TYPE=A or E (ASCII or EBCDIC), and is not being scanned for records. A command was detected but it had no record, that is, another command or $$END. (This scenario is very similar to that involving Reason Code 1.) The SITE or LOCSITE command was used with the RIFS=NO option to change the scan processing. The RIFS (Recordize Input File Structure) option changes the incoming batch to a record structure and recognizes carriage returns and line feeds (CRLF) in ASCII data and new line feeds (NL) in EBCDIC data. For more information on the RIFS parameter, see *Processing $$ADD Commands Embedded in Batches* on page 103.
4	/*BINASC detected in the current record.	A /*BINASC command tells the scanner that the rest of the transmission is binary data., which requires that scanning be turned off.
5	$$ADD command with SCAN=NO was detected	The $$ADD parameter requested that scanning be turned off.
6	$$ADD command without SCAN=YES was detected and SCAN=ALL wasn't the initial setting	The initial setting of SCAN=YES requires that each $$ADD command have the SCAN=YES parameter for scanning to continue from that point on.

Appendix A

Automating Connect:Enterprise

This chapter describes how to automate Connect:Enterprise by using the following features:

- The End-of-Batch exit (EOBX)
- The Connect:Enterprise Modify feature
- The USERRCD and AUTOSEND options on EXTRACT and ADD utilities
- The Connect:Enterprise Auto Connect Time parameter
- Application agents
- Integrating the Connect:Enterprise automation features

This chapter concludes with two examples that illustrate Connect:Enterprise automation.

The End-of-Batch Exit

The End-of-Batch exit enables you to perform an automatic function at the successful end of an online ADD collection. Information about the new batch, such as Mailbox ID, batch number, and number of blocks in the batch, is available to the exit program upon entry. From this data, the End-of-Batch exit determines what action to take. Although the user exit could perform a number of functions, typically a job is submitted to perform the function.

The STEOBX and STEOBX2 sample programs provided with Connect:Enterprise illustrate the process of:

- Examining the input data, such as Mailbox ID
- Comparing the data against a table
- Opening the JES reader and writing a file to JES to issue a command or run a job

You can submit custom JCL tailored to this batch, or issue console commands using the //F PROCSTEP,.... JCL. Connect:Enterprise must be a STARTED TASK running with the option MODIFY=YES to accept MODIFY commands through this method.

See the *Connect:Enterprise for z/OS Application Agents and User Exits Guide* for more information about the End-of-Batch exit.

Connect:Enterprise Modify Feature

The Connect:Enterprise Modify feature replaces the Connect:Enterprise Write To Operator With Reply (WTOR) interface when MODIFY=YES is coded in the *OPTIONS section of Connect:Enterprise. The Modify feature works the same as the WTOR interface except that a MODIFY command is used instead of a reply number. This provides an advantage when automating a Connect:Enterprise function because you do not need to know the reply number.

To use the Connect:Enterprise Modify feature, Connect:Enterprise must be executed as a *started task*. The same MODIFY command can be stored and used repeatedly.

The End-of-Batch exit and the AUTOSEND function on the ADD utility and application agents can best use this feature.

USERRCD and AUTOSEND Options on EXTRACT and ADD Utilities

The ADD and EXTRACT offline utilities have a USERRCD feature. The USERRCD option enables you to construct a single record of data to be inserted as the first record of the batch. The record is made from user control information input, and allows for symbolic substitution of the following:

- Time
- Date
- Batch number
- User batch ID
- Mailbox ID

This is the first record written to the Connect:Enterprise VSAM batch queue when the ADD utility is used with USERRCD. When invoked by EXTRACT, the user record is the first record written to the output file.

The AUTOSEND option is available on the ADD utility only. This feature provides symbolic substitution and sends the information provided to the JES internal reader. You can use the AUTOSEND function to issue a modify command to start an Auto Connect list, freeing the operator of this task.

You may also choose to issue and modify commands to invoke the end-of-batch application agent through a $$INVOKE. This allows you to centralize all offline and online data collection automation. If $$INVOKE is used, the symbolic substitution of variables contained in USERRCD is not performed prior to INTERNAL READER.

LOG=YES Option on EXTRACT and ADD Utilities

If you have an automation or scheduler system that can read an input file and take action based on data in the file, consider using the ADD and EXTRACT logging done when LOG=YES is specified.

ADD and EXTRACT log records can be written into a sequential file specified by the LOGFILE DD statement in the offline utility JCL. Your automation or scheduler can monitor the data set for updates and take action based on the newly written records.

Ensure that all offline utility job streams use the LOGFILE data set. Allocate the data set with DISP=SHR. New records are always written to the end of the file.

Connect:Enterprise Auto Connect Time Parameter

The time parameter in the Auto Connect list automatically starts an Auto Connect list at one or more specified times. This feature enables you to automatically send the same batches, or group of batches, to any set of remote sites at one or more prearranged times. Use the *CALENDAR section of the ODF to select or limit the date or days for a time-initiated Auto Connect session.

Application Agents

Application agents enable Connect:Enterprise to automatically react to events as they occur. By coding a simple set of rules, you can define the selection criteria and actions that can be initiated when events occur. Actions that can be initiated are:

- Submit batch jobs
- Issue console commands
- Issue console messages
- Send files to Connect:Direct
- Call user programs
- Issue SNMP traps

See the *Connect:Enterprise for z/OS for z/OS Application Agents and User Exits Guide* for more information.

Integrating the Connect:Enterprise Automation Features

When the Connect:Enterprise automated operation features described previously are integrated, they provide a powerful tool that can connect Connect:Enterprise sessions anywhere across a network, or automate the processing of data when received. If you know in advance where data is intended to go, such as processing or distribution, you can get it there without operator intervention.

Sample Automation Implementations

The following are sample automation implementations that can assist you in understanding how to automate Connect:Enterprise. Both show a way to implement the End-of-Batch exit to automate processing. This could also be implemented using other Connect:Enterprise exits or application agents. The best solution depends on specific site requirements.

Example 1

This example shows how Connect:Enterprise can schedule batch and data transmission activity based on data arrival. This activity enables Connect:Enterprise to automate the data flow.

In this example, a bank customer sends automated clearinghouse data to the bank mainframe in San Francisco using a local connection to the remote data center in Los Angeles. The customer requires an acknowledgment of receipt of the data and verification that the data was valid. The bank operator is not required for normal processing.

1. The customer dials the bank's remote site in Los Angeles and the call is answered automatically. The customer's data is identified with $$ADD as the first record of data. Connect:Enterprise L.A. drives the bank's End-of-Batch exit when the transmission is completed without errors.
2. The End-of-Batch exit submits the customized JCL to:
   - Extract the data.
   - Preprocess the data.
   - Invoke Connect:Tracs to send the data to the bank's Connect:Enterprise San Francisco central site location (if the data was valid) using the SNA network.
   - If the data is in error, a special report is generated and transmission indicating the error is automatically sent to the customer. If the data is good, a confirmation notice is sent to the customer.
3. At the bank's central site in San Francisco, Connect:Enterprise is running an End-of-Batch exit also. When the data is successfully received, the central site Connect:Enterprise submits the JCL to accumulate the data for later processing.

The key to this process is coordinating the ODFs, the End-of-Batch exits, and the identification of the batches, such as Mailbox ID and batch ID.

Coding the End-of-Batch Exit and ODF for Los Angeles

The trigger for the automated process is the completion of the batch collection. The batch must be identified as a particular unit of work for a specific process to occur. The remote site makes this identification with the ID= and BATCHID= parameters as shown:

```
$$ADD ID=REMOTE1 BATCHID='REMOTE1 ACH DATA'
DATA FILE INSERTED HERE *******
```

In addition to the usual remote definitions that Connect:Enterprise requires to communicate with REMOTE1, Connect:Enterprise needs to know the module name for the End-of-Batch exit and the Listname information in order for the remote to receive the confirmation notice.

The L.A. Connect:Enterprise uses the following ODF:

```
*OPTIONS
 VPF='ENTPRS.VPF'
  VTAM=YES
  BTAM=YES
  CONSLOG=YES
  XENDOFB=STEOBX2
  MODIFY=YES
  APPLID=CMBOXLA
  PASSWORD=PASSWRD
  VSESSLIM=9
  UA=USERASM
  LOGONMSG='SUCCESSFUL LOGON TO Connect:Enterprise'
*CONNECT
  LISTNAME=REMOTE1
    TYPE=BSCAD
      REMOTE1 07 5551212 MODE=SENDONLY
  LISTNAME=REMOTE2
    TYPE=LU1RJE
      REMOTE2 MEDIA=PU
*REMOTES
  NAME=REMOTE2
    TYPE=LU1RJE
      LUNAME=TRCLU01
      CONSOLE=YES
```

STEOBX2 was assembled and linked into the Connect:Enterprise LOADLIB from STEOBX2 located in the source file shipped with Connect:Enterprise.

The sample exit STEOBX2 in its original form requires access to a JES internal reader and a single PDS. The PDS contains a SUBTABLE member which contains a table of jobs to be submitted. The record format is:

```
x yyyyyyyy zzzzzzzzzzzzzzzzzzzzzzzz jjjjjjjj
```

The parameters are described in the following table:

Parameter	Description
x	The match code. Starts in column 1. The values are: 1 = ID only, 2 = BATCHID only 3 = BOTH ID and BATCHID must match
y	The Mailbox ID to match against the incoming batch. Starts in column 3.
z	The BATCHID to be matched against, without beginning or ending quotation marks. Starts in record column 12.
j	The member name of the JCL to be read into the internal reader from this PDS after the batch is successfully received. Starts in column 37.

The following JCL runs the remote Connect:Enterprise as a started task and invokes the End-of-Batch exit:

```
//CMBOXLA  PROC  MEMBER=’OPTDEF1’
//CMBOXLA  EXEC  PGM=STMAIN,PARM=’SRV1’,REGION=4500K,TIME=1440
//STEPLIB   DD    DISP=SHR,DSN=ENTPRS.LOAD
//SYSUDUMP  DD    SYSOUT=*
//SYSPRINT  DD    SYSOUT=*
//BTSNAP    DD    SYSOUT=*,DCB=(LRECL=125,BLKSIZE=1632,RECFM=VBA)
//*
//*END-OF-BATCH DD CARDS FOLLOW
//*
//STINTRDR  DD    SYSOUT=*
//STPDS     DD    DSN=ENTPRS.SUBTABLE(SUBTABLE),DISP=SHR
//*
//SNAPOUT   DD    SYSOUT=*,DCB=(LRECL=125,BLKSIZE=1632,RECFM=VBA)
//BSCNS     DD    UNIT=001
//BSCSW     DD    UNIT=002
//OPTDEF    DD    DSN=ENTPRS.OPTFILE(&MEMBER),DISP=SHR
```

Coding the EXTRACT for the Los Angeles Site

The following model JCL was submitted by the End-of-Batch exit to extract the batch received, preprocess it, transmit a response to the customer, and transmit the data to the central Connect:Enterprise site in San Francisco if the data received is valid.

Note: Remember that the End-of-Batch exit checks the SUBTABLE member of the PDS referenced by the //STPDS DD that was used by Connect:Enterprise for a match. The member name referenced on the matching table entry contains the JCL that is submitted.

```
EXTRACT / Connect:Tracs for MVS JCL Log Angeles
//USERJOB JOB (ACCOUNT INFO),JOBCARD INFO
//*******************************************************
//* EXTRACT A BATCH FROM THE Connect:Enterprise VSAM BATCH FILES
//*******************************************************
//EXTRACT EXEC PGM=STOUTL,PARM='SRV1',REGION=4000K
//STEPLIB   DD  DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//BTSNAP    DD  SYSOUT=*,DCB=(REFCM=FBA,LRECL=133,BLKSIZE=266)
//REPORTS   DD  SYSOUT=*,DCB=(REFCM=FBA,LRECL=133,BLKSIZE=266)
//PRINT     DD  SYSOUT=*,DCB=(REFCM=FBA,LRECL=133,BLKSIZE=266)
//SYSTERM   DD  SYSOUT=*,DCB=(REFCM=FBA,LRECL=133,BLKSIZE=266)
//OUTFILE   DD  DSN=&&TEMP1,DCB=(LRECL=94,BLKSIZE=9400,RECFM=FB),
//              UNIT=3380,SPACE=(TRK,(5,1)),DISP=(,PASS)
//SYSIN     DD *
  EXTRACT
  VPF='ENTPRS.VPF'
  DELETE=YES
  ID=REMOTE1
  USERRCD=1
$$ADD ID=&IDFIELD BATCHID='PREPROCESSED REMOTE1 ACH DATA'
//****************************************************
//* PREPROCESS THE DATA, CREATE CONFIRMATION NOTICE
//* SET CONDITION CODE IF BAD DATA DETECTED
//****************************************************
//PROCESS EXEC  PGM=USER,REGION=1024K
//STEPLIB   DD  DISP=SHR,DSN=USER.APPL.LOADLIB
//INFILE    DD  DSN=&&TEMP1,DISP=SHR
//CUSTFILE  DD  DSN=&&TEMP2,DISP=(,PASS),SPACE=????,UNIT=????
//OPTDEF    DD *
//*************************************************
//* TRACS - EXEC PGM=TRCSNA Connect:Tracs TO Connect:Enterprise
//*************************************************
//TRACS   EXEC  PGM=TRCSNA,REGION=1024K,COND=(0,LT)
//STEPLIB   DD  DISP=SHR,DSN=USER.TRACSSNA.LOADLIB
//OPTDEF    DD  *
```

Note: The Connect:Tracs Log mode entry was not from the IBM default mode table. Instead, it was assembled from the *Connect:Tracs for MVS (SNA) User's Reference Manual* or the *Connect:Tracs for VSE (SNA) User's Reference Manual*.

```
*OPTIONS  THIS CONNECT:TRACS IS THE SLU TALKING TO Connect:Enterprise
APPLID=TRACSNA,
PASSWORD=PASSWRD,
COMPRESS=YES,
DISCINTV=0005,
OPMODE=AUTO,
RMTAPPLID=CMBOXSF,
RMTDEVICE=CMBOX,
RCVBUFF=512,
SNDBUFF=512,
USERDATA=TRACSLA,
LOGMOD=RJE3770
//SNDFILE   DD  DSN=&&TEMP1,DISP=SHR
//RCVFILE   DD  DUMMY,DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUDUMP  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SNAPOUT   DD  SYSOUT=*
//************************************************************
//*       ADD THE CUSTOMER RESPONSE BATCH TO Connect:Enterprise
//************************************************************
//ADD     EXEC  PGM=STOUTL,PARM=’SRV1’,REGION=4000K
//STEPLIB   DD  DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//BTSNAP    DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//REPORTS   DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//SYSTERM   DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=266)
//INFILE    DD  DSN=USER.APPL.RECEIPT,DISP=SHR
//SYSIN     DD  DATA,DLM=ZZ
  ADD
  VPF=’ENTPRS.VPF’
  ID=REMOTE
  BID=’REMOTE1 ACH CONFIRMATION’
  AUTOSEND=3,E
//USERJOB JOB (ACCOUNT DATA),’AUTOSEND JOB’,CLASS=X,MSGCLASS=A
// F CMBOXLA,$$CONNECT L=REMOTE1
//STEP1  EXEC  PGM=IEFBR14
ZZ
```

Coding the Options Definition File for San Francisco

Connect:Enterprise is running in San Francisco as a started task, with similar JCL to Los Angeles. The same End-of-Batch exit is running but with a different table. The Connect:Tracs remote is defined to Connect:Enterprise in the *REMOTES section. The central site Connect:Enterprise uses the following ODF:

```
*OPTIONS
  VTAM=YES
  VPF='ENTPRS.VPF'
  BTAM=YES
  CONSLOG=YES
  XENDOFB=STEOBX2
  MODIFY=YES
  APPLID=CMBOXSF
  PASSWORD=PASSWRD
  VSESSLIM=9
  UA=USERASM
  LOGONMSG='SUCCESSFUL LOGON TO Connect:Enterprise'
*CONNECT
  LISTNAME=REMOTEX
    TYPE=BSCAD
      REMOTE1 07 5551212 MODE=SENDONLY
  LISTNAME=REMOTEY
    TYPE=LU1RJE
      REMOTEY MEDIA=PU
*REMOTES
  NAME=TRACSLA
    TYPE=LU1RJE
      LUNAME=TRACSNA
      CONSOLE=YES
      COMPRESS=YES
  NAME=REMOTEY
    TYPE=LU1RJE
      LUNAME=TRCLU101 TRCLU102 TRCLU103 TRCLU104
      CONSOLE=YES
      COMPRESS=YES
      DISCINTV=010
```

Coding the EXTRACT for San Francisco

The following EXTRACT JOB was submitted by the End-of-Batch exit in San Francisco:

```
//USERJOB JOB (ACCOUNT INFO),JOBCARD INFO
//**********************************************************
//*       EXTRACT A BATCH FROM THE Connect:Enterprise VSAM BATCH FILES
//**********************************************************
//EXTRACT EXEC PGM=STOUTL,PARM=’SRV1’,REGION=4000K
//STEPLIB   DD DSN=ENTPRS.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*,
//PRINT     DD SYSOUT=*,
//SYSTERM   DD SYSOUT=*,
//BTSNAP    DD SYSOUT=*,
//REPORTS   DD SYSOUT=*,
//OUTFILE   DD DSN=USER.FILE1(+1),DISP=(,CATLG,DELETE),
//             DCB=(LRECL=94,BLKSIZE=9400,RECFM=FB),
//             UNIT=3380,SPACE=(TRK,(5,1))
//SYSIN   DD /DATA,DLM=ZZ
  EXTRACT
  DELETE=YES
  ID=TRACSNA
  VPF=’ENTPRS.VPF’
  ZZ
```

The batch is flagged as deleted when it is extracted. This flag prevents the batch from being extracted a second time if another batch arrives with the same ID and BATCHID.

Example 2

This example shows how the scheduling of Connect:Tracs by Connect:Enterprise allows data to flow unattended across an SNA network to another Connect:Enterprise system, which in turn undertakes a new action based on an End-of-Batch exit.

This example includes a vendor shipping information to all branch offices with a single transmission to a single data center.

1. The vendor initiates the action by sending the data to Los Angeles.
2. Connect:Enterprise in Los Angeles has an End-of-Batch exit coded. The BATCHID of the transmission matches one in the End-of-Batch exit BATCHID Table.
3. The End-of-Batch exit submits the JCL to JES, which issues a modify command to the console-started Connect:Enterprise system in Los Angeles, invoking an Auto Connect session to send the file to all branch offices located close to the Los Angeles data center.
4. Then the same JOB extracts the data and invokes Connect:Tracs to send the data to the Boston and Detroit data centers.
5. In Boston and Detroit, the Connect:Enterprise systems have End-of-Batch exit. The BATCHID of the Los Angeles transmission matches an entry in their BATCHID tables and the JCL is submitted to JES. A modify command is issued to the console-started Connect:Enterprise system, invoking an Auto Connect list or lists to distribute the file to all branch offices close to these data centers.

Appendix B

Worksheet for Remote Sites

The *Worksheet for Remote Sites* specifies the information that remote sites need to initiate connections to the Connect:Enterprise for z/OS repository. Most of this information must be provided by the host site administrator for use by remote site operators.

Worksheet for Remote Sites	
Connection Information	
Hours of access to Connect:Enterprise for z/OS	______________________
Terminal options	Specify any application-determined options, automatic logon capability, or program boot options required for Connect:Enterprise. ______________________ ______________________ ______________________ ______________________ ______________________ ______________________
SNA logon to Connect:Enterprise	A remote site must issue the proper logon to Connect:Enterprise. The logon command format depends on the capabilities of the remote device and on the generation of certain VTAM tables at the host site. Connect:Enterprise requires certain information before it accepts the logon, but the host-site VTAM table generation supplies the correct information. Logon to Connect:Enterprise can be similar to the following: LOGON APPLID(MAILBOX) LOGMODE(RJE3770) DATA(RMTxxx) ____
IP address (FTP only)	Specify the IP address of the host Connect:Enterprise system. ____
Port Number (FTP only)	Specify the port number that Connect:Enterprise monitors for connections. Default = 5555__________________
User ID (FTP only)	The user ID to specify during logon; remote name defined in the ODF *REMOTES record or ANONYMOUS. See Chapter 5, *Configuring ODF Records for FTP Connections*, for more information. ______________________
Password (FTP only)	The password assigned to the remote name. This is required for FTP connections if security checking is activated through the security interface. ______________________
SIGNON data format (BSC only)	Remote sites that communicate with Connect:Enterprise through JES or POWER may need to supply a SIGNON record when an initial connection is established. Also, BSC sites can supply a free-form BSC signon to Connect:Enterprise providing a remote name and password for remote site identification or security checking. If BSC SIGNON records are defined in the ODF, and the remote site supplies a signon, tell the operators the exact signon format. ______________________
Connection Information	
Connect:Enterprise password access	Either list the current password or indicate the host-site personnel to contact to obtain the password. Optionally, indicate that the remote site cannot obtain the Connect:Enterprise password. ______________________

Worksheet for Remote Sites	
Switched line phone numbers	List one or more phone numbers to use when dialing Connect:Enterprise on switched lines. __________ __________
FTP Security Information	
Using SSL or TLS protocol?	Create key database. See Chapter 5, *Configuring ODF Records for FTP Connections*. key label: __________ path to key database: __________ file name of key database: __________ password generated for key database: __________
Mailbox Batch Information	
Connect:Enterprise Mailbox IDs	List one or more Mailbox IDs used by the remote site operators to identify Connect:Enterprise batches. This is the 1–8 character ID. __________ __________ __________ __________ __________ __________
Connect:Enterprise batch IDs	List one or more user batch IDs used by the remote site operators, or identify generalized standards for the user batch IDs in your system. This is the 1–64 character BATCHID. __________ __________ __________ __________ __________ __________
Contact Information for Connect:Enterprise Support	
Phone numbers of host site support personnel	__________ __________ __________
Instructions to access Connect:Enterprise	
	List the step-by-step instructions the remote site operator must perform to access Connect:Enterprise. This can include dialing information and modem use if the remote site access is through a switched line. __________ __________ __________ __________ __________ __________ __________ __________ __________ __________ __________ __________

Glossary

A

ACQUEUE

Specifies the disposition of an Auto Connect session that is unable to be initiated because there is no BSC line, SNA session, or FTP thread available or the Auto Connect session is currently active. When the parameter ACQUEUE=YES is specified, the Auto Connect session is queued and initiation is attempted at a later time. Otherwise, the Auto Connect session is terminated with an error condition.

ADD Utility

A set of instructions used to submit the Connect:Enterprise ADD utility. The ADD utility is used to add batches to the VSAM batch files for access by the remote sites.

APPL (Application)

See *VTAM Application Program*.

Application Agent

A Connect:Enterprise interface that allows the customization of Connect:Enterprise execution. Each application agent is driven by a user-defined set of rules. The rules can display system console messages, issue system console commands, execute programs, and submit jobs. Connect:Enterprise supports the following application agents: Console, End Of Batch, Logging, Scheduler, and Wake Up Terminate.

Auto Connect

A Connect:Enterprise feature that allows host-initiated data communications to one or more remote sites. The host and remote sites may be connected using SNA, FTP, bisync manual dial, auto dial, or nonswitched lines. The Auto Connect session may be fully automated by time of day, or controlled with the $$CONNECT console command. Full reporting of Auto Connect activity is available.

Auto Dial

Refers to the capability of the host computer to automatically dial the remote site to establish a connection on a switched line. The Auto Dial feature is usually generated for the Transmission Control Unit or front-end processor of the host site on a line-by-line basis.

B

Batch

A set of related data collected by or added to Connect:Enterprise and maintained on the VSAM Batch Files at the host.

Batch Number

A unique 7-digit number assigned internally by Connect:Enterprise to each individual batch on the VSAM Batch Files. The number may be obtained by the $$DIRECTORY function or the offline utilities LIST function.

Batch Queue

See *VBQ (VSAM Batch Queue)*.

Batch Security

Optional Connect:Enterprise method of providing security for remote site access to the system. Mailbox IDs are assigned to remote sites and defined as valid at the host site. If Batch Security is used, remote sites must supply a valid ID as part of the $$ commands that access the Connect:Enterprise data files. (Formerly called ID Validation.)

Batch Status

A set of flags maintained for each batch on the VSAM Batch Files. The Batch Status flags are displayed in the LIST offline utility report or the $$DIRECTORY output data. Some of the Batch Status indicators are incomplete batch, deleted batch, batch transmitted to remote site, and batch extracted at the host site.

Batch Type

Used to indicate which batches to recall from Connect:Enterprise. Types include batches containing data received from remote sites and batches containing data to be transmitted.

Blank Compression

A method of replacing strings of contiguous blanks with control characters indicating the number of blanks removed. Commonly used to shorten the amount of data sent over telecommunications lines. Connect:Enterprise uses standard 3780 blank compression techniques on BSC lines and standard SNA blank and character compression on SNA sessions.

Blank Truncation

A method of dropping trailing blanks from the end of fixed length data records before sending the data over telecommunications lines. Used by Connect:Enterprise as an option to shorten the amount of data sent over telecommunications lines.

BSC (Binary Synchronous)

A standard telecommunications line protocol used to transmit blocks of data over telecommunications lines between host computers and remote sites. Binary Synchronous (also known as bisync) allows a faster transmission rate than a start/stop protocol, because its ratio of data bits to checking bits is higher. This line protocol is used by Connect:Enterprise.

BTAM (Basic Telecommunications Access Method)

A standard IBM access method used by Connect:Enterprise to read and write data over telecommunications lines to a variety of terminals and devices.

BTAM ID Verification

An optional BTAM feature that enables the exchange and verification of host site and remote site IDs. Available on switched lines only, the feature provides added security in a Connect:Enterprise system. Both the host site and the remote site must be capable of implementing the option. Connect:Enterprise allows the host site ID to be sent, the remote site ID to be received, or both IDs to be exchanged.

C

Clear Control Channel (CCC)

A command that enables Connect:Enterprise to negotiate a clear-text control channel after the user ID and password have been transmitted in encrypted format. The control channel remains in clear-text until the connection ends. All data and objects transferred between the client and server remain encrypted. Both ends of the connection must support the use of this command.

Compression

See *Blank Compression*.

Connection ID

The CICS definition that describes the remote system in terms of Netname (APPLID). The connection ID is a local name (within the local CICS only) that is used to define the remote partner system (Connect:Enterprise).

Cross System Client Utility (CSCU)

A Connect:Enterprise utility that enables you to use a subset of the offline utilities to access the VSAM batch and log files from a remote logical partitioning (LPAR), unlike offline utilities which must run from the same LPAR as the Connect:Enterprise VSAM File Server. CSCU control and output is similar to the offline utilities.

D

Data Collection

The process in which Connect:Enterprise collects data from remote sites and stores it in the VSAM Batch Files. Data Collection means data is input from a remote site to Connect:Enterprise at the host computer.

Data Repository

The component that transmits and collects data from BSC, FTP, and SNA sites. The repository handles all session activity and accepts service requests from the console, the user API, the ISPF interface, the CICS interface, and the Connect:Enterprise FTP server.

Data Transmission

The process in which Connect:Enterprise transmits data from the VSAM Batch Files to remote sites. Data transmission means data is output from Connect:Enterprise at the host computer to the remote site.

Directory

A formatted listing of control information for batches on the Connect:Enterprise VSAM Batch Files. It is obtained from the $$DIRECTORY command.

Disconnect Interval

The number of seconds a session may be inactive before forcing session termination. This may differ for each remote site defined to Connect:Enterprise. This safety feature, which is implemented using the DISCINTV parameter, is used to reduce the use of resources by remote sites that have no current activity and to prevent an Auto Connect session from suspending if a remote site does not respond.

EXTRACT Utility Model

A set of JCL statements and parameter (specification) data submitted by Connect:Enterprise CICS or ISPF interface to initiate execution of the Connect:Enterprise EXTRACT utility. The EXTRACT utility is used to retrieve batches from VSAM batch files to a sequential output file.

F

FMH (Function Management Header)

A standard SNA feature that allows a data stream to be sent to a specific destination and controls the way the data is presented at the destination. Connect:Enterprise supports FMH Type 1, a 6-character field sent at the start and the end of a data stream. This FMH selects the media used for the data, marks the beginning and end of a Connect:Enterprise batch, and further describes the format of the data.

FTP (File Transfer Protocol)

An international standard for reading and writing files across a TCP/IP network.

FTP Server

The capability of Connect:Enterprise to function as an FTP server. This enables remote FTP client sites to access, retrieve, and send data to the Connect:Enterprise batch queues through standard FTP commands.

G

GSKKYMAN

An IBM utility that is used to create and maintain the SSL key database.

H

Host

The main processing computer where Connect:Enterprise is running and where you send your data batches. Also referred to as the host site or host computer.

I

IRS (Inter-Record Separator)

A special character used to separate multiple records in a block of data being transmitted over a telecommunications line. Connect:Enterprise allows either X'1E' or X'1F' as the inter-record separator on BSC lines, and allows only X'1E' for SNA sessions. Also referred to as an IRS.

J

Job Entry Subsystem (JES)

A system facility for spooling, job queuing, and managing job-related data.

L

Leased Line

Refers to telecommunications lines on which connection is not established through a switched network. Connect:Enterprise Leased Line support is point-to-point and therefore allows data to be exchanged only between the host site and a single remote site. Leased Multipoint lines are not supported by BSC connections in Connect:Enterprise.

Line ID

Uniquely identifies a BSC line that is accessed during Auto and Remote Connects. This is a BSC-only entry generated by a nonswitched M$LINE or M$LINEX macro in the User Assembly.

List Name

The Auto Connect List Name defined in the Connect:Enterprise ODF.

Log Facility

A Connect:Enterprise feature that provides file logging and full reporting for remote-initiated transactions. An additional option provides host system console log messages both for host-initiated and for remote-initiated connections and disconnections.

LOGOFF

The process of ending a remote site session with a host site program such as Connect:Enterprise. A LOGOFF may be a text command or a control function from a remote device.

LOGON

The process of establishing a session between a remote site and a host site program such as Connect:Enterprise. A LOGON may be automatic after a connection is established, or may be entered as a text command or a control function. In Connect:Enterprise, either the remote site or the host site may attempt to initiate the LOGON process.

Logon Mode Table

A table defined to VTAM containing a set of entries that provide session parameters, or the rules for controlling SNA communications. The LOGON that attempts to establish a session causes access to this table to obtain the session rules.

LOGON Security

An optional Connect:Enterprise/SNA method of providing security during a remote site’s attempt to LOGON to Connect:Enterprise. The LUNAME (assigned to the remote site as part of the VTAM definition process) is provided to and validated by Connect:Enterprise when a LOGON is attempted.

LU (Logical Unit)

A logical unit provides the port for user access to an SNA network. Each remote device that can establish a session with Connect:Enterprise is a logical unit.

LU1RJE (LU Type 1 RJE)

A device emulating 3770, or a similar device or software package that uses Logical Unit Type 1 protocols and is used primarily for data transfer or RJE (Remote Job Entry) purposes. The devices typically have multiple I/O devices, such as printers, card readers, and storage devices. An operator console for messages or interactive use is often present.

M

Mailbox ID

The 1–8 character ID which defines batches in the VSAM Batch Files.

Mailbox Name

The 8-character symbolic name used to identify individual Connect:Enterprise systems to the user interface.

Mailbox Password

A security password used to control access to Connect:Enterprise systems.

Mailbox User ID

An 8-character field used to identify each user to Connect:Enterprise. In order for a user to access a Connect:Enterprise system, the User ID must be defined and assigned. The CICS and ISPF Interface panel displays the current user in the upper right corner.

Manual Dial

Refers to the method the host site uses to dial remote sites to establish a connection on a switched line. With Manual Dial, an operator at the host site must manually dial the telephone number of the remote site if the connection is initiated by the host site.

If the connection is initiated by the remote site, the manual dialing at the host is not used.

Media

An input/output device on a terminal, such as a printer, card reader, card punch, keyboard, display, or diskette. Commonly available on LU Type 1 RJE terminals, and supported by Connect:Enterprise/SNA.

MLU (Multiple Logical Unit)

A terminal designed to allow the operation of more than one session between a remote terminal and a host site such as Connect:Enterprise. A single terminal may actually appear as multiple devices, and may have concurrent inbound and outbound data streams active for each. Some 3770-type devices have this capability. Connect:Enterprise supports up to six MLU sessions per remote site.

N

NCP (Network Control Program)

The Network Control Program, generated by host site personnel, that controls the operations of a communications controller such as a 37x5.

Non-Switched Line

A telecommunications line on which connection is not established through a switched network. Sometimes referred to as a Leased Line.

NPSI (Network Control Program Packet Switching Interface)

An IBM licensed program that allows SNA users to communicate over packet switching data networks that have interfaces complying with CCITT Recommendation X.25. It allows SNA programs to communicate with SNA or non-SNA equipment over such networks.

O

(ODF) Options Definition File

A file containing Connect:Enterprise control records and keyword parameters that specify options in effect for the current execution of online Connect:Enterprise. The file contains options that control security, password, Auto Dial telephone numbers, SIGNON records, Auto Connect, SNA sites, and other system options.

Offline Utilities

The Connect:Enterprise utilities used to access and maintain the data batches on the VSAM Batch Files. The offline utilities allow you to LIST control information for batches, ADD batches, EXTRACT batches, DELETE batches, ERASE batches, alter batch status flags (STATFLG), MOVE batches from one VBQ to another, and REPORT on session activity.

P

Password

See *Mailbox Password*.

PLU (Primary Logical Unit)

In a particular session between two LUs, one LU adheres to a set of SNA-defined primary protocols and is known as the primary logical unit (PLU) for that session. The other LU adheres to a set of secondary protocols and is known as the secondary logical unit (SLU) for that session. More than one session can exist between two LUs. Multiple concurrent sessions between the same two LUs are referred to as parallel sessions. Not all LUs have parallel session capability.

Point-to-Point Line

A telecommunications line connection that allows data exchange between two points on the connection, usually the host site and a remote site. When a dialed connection is established on a switched network, the connection is considered point-to-point. Leased lines where the remote site is a single station are also considered point-to-point.

R

RDW (Record Descriptor Word)

A 4-byte field used to define the length of variable length records within a data file. For batch data coming into Connect:Enterprise (ADD), the RDW may be removed or retained. For batch data sent from Connect:Enterprise (REQUEST) the RDW may be created or not created.

Remote Name

A 1–8 character name assigned to identify a remote site that may be contacted by the host site during an Auto Connect session. Also used to identify every remote site that can establish a session with Connect:Enterprise.

Remote Site

Any terminal, computer, or software that can connect with Connect:Enterprise in the host computer.

REXX (Restructured Extended Executor) Language

A general-purpose, procedural language for scripting end-user programs designed for IBM systems.

RFC (Request for Comments)

One of a series, begun in 1969, of numbered Internet informational documents and standards widely followed by commercial software and freeware in the Internet and UNIX communities.

S

Session

A logical connection between Connect:Enterprise at the host site and another logical unit, such as a 3770 device. When a LOGON is completed between Connect:Enterprise and a remote site, they are said to be in session.

SIGNON

A special format data record sent by some remote BSC terminals designed to communicate with RJE software (such as JES or VSE POWER) in the host computer. The SIGNON record may be required by Connect:Enterprise provided Connect:Enterprise has been configured to do so when installed. The SIGNON format(s) used must also be specified at installation. A SIGNON is not required and not supported for SNA remote sites.

SLU (Secondary Logical Unit)

See *PLU (Primary Logical Unit)*.

SNA (Systems Network Architecture)

A set of rules, procedures, and structures for a communications network.

Socket Number

A two way connection identified by the unique combination of IP addresses and port numbers in a given connection. For example, the following combination illustrates the unique ID representing a complete socket: Client IPAddress/Port Number - Server IPAddress/Port Number.

SPLITCOUNT

Specifies a 1–4 digit numeric count of records to be contained in an added batch, allowing you to split a large sequential input file into several smaller batches with the same batch identifiers. Sequential input records are read and added to the output batch until the SPLITCOUNT limit is reached. Connect:Enterprise then closes out the batch and begins a new batch with the same identifiers.

SSL (Secure Sockets Layer)

A protocol for transmitting private documents over the Internet. SSL uses a private key to encrypt data that is transferred over the SSL connection.

Status Codes

The status flag indicators for a batch. Codes include the following: D, deleted; T, transmitted; R, Requestable; E, Extracted; M, Multxmit (for a list of these codes, see information on VSAM Batch Status Flags in the *Connect:Enterprise for z/OS User's Guide*).

Switched Line

A telecommunications line on which connection is established over a switched (dialup) telephone line.

T

TLS (Transport Layer Security)

A protocol based on SSL 3.0 protocol specification and designed to provide privacy and data integrity between two communicating applications.

TRACE

In Connect:Enterprise, the capability to create a snapshot dump of internal Connect:Enterprise control information for communications activity, User Exit calls, or VSAM Batch Files access.

Transparency

A method of transmitting data over a telecommunications line wherein special line control characters embedded in the data are transparent and do not function in their normal capacity as line control characters. Transparency is used when non-text data (such as object modules or other binary data) must be sent over telecommunications lines. Connect:Enterprise supports both BSC transparency and SNA transparency.

Truncation

See *Blank Truncation*.

$TURNLINE$

An optional feature in Connect:Enterprise that provides for a limited conversational mode transmission. When a $TURNLINE$ record is encountered in data being sent to a remote site, the sender temporarily stops sending and issues the proper BSC protocol to turn around the line and begin receiving. After all data is received, sending resumes with the record following $TURNLINE$.

U

User

See *Mailbox User ID*.

User Assembly

A series of macros used to define a network of BSC lines to be used by Connect:Enterprise. The macros are generated by each user to define their requirements and input to the Assembler to create a module for use by Connect:Enterprise BSC connections. A User Assembly is not required by SNA connections.

User Batch ID

A 1–64 character free-form batch identifier used to describe the contents of a batch of data on the Connect:Enterprise VSAM Batch Files.

User Exits

A user-written program called by online Connect:Enterprise, offline utilities, and the CICS interface at appropriate times during the processing of a transaction. The user-supplied program can thereby alter the standard processing done by Connect:Enterprise. User Exits may be supplied to examine all input data from a remote site, to examine output data to a remote site, to provide unique security processing, or to examine and alter data in Connect:Enterprise $$ commands. No alteration of data is possible by a user exit in the offline utilities and the CICS interface processing.

USS Table

A table defined to VTAM that provides conversion of character-coded LOGON or LOGOFF to field-formatted LOGON or LOGOFF. You may need to provide this table to VTAM to allow a remote site to establish and terminate SNA sessions with Connect:Enterprise.

V

VBQ (VSAM Batch Queue)

The Connect:Enterprise data set used for storing batches of data collected from remote sites during online Connect:Enterprise. These batches may be available for transmission to remote sites, and are always available for extraction at the host site. The VSAM Batch Queue may be defined as a single VSAM cluster or up to 20 VSAM clusters that are processed as a single repository for batch data. The VSAM Batch Queue contains multiple individual batches of data which can be accessed by their Mailbox ID.

VBQ Blocking

A Connect:Enterprise feature that blocks multiple records or collection buffers into a single VBQ record for transmission. This improves transmission performance by reducing the disk I/O overhead.

VCF (VSAM Control File)

The Connect:Enterprise data set that contains control information for batches stored on the VSAM Batch Queue.

VLF (VSAM Log File)

The Connect:Enterprise data set that contains logged information on the progress of a Connect:Enterprise execution.

VPF (VSAM Pointer File)

The Connect:Enterprise data set that contains control information for every file defined in the Connect:Enterprise system and locator information for every existing batch.

VSAM (Virtual Storage Access Method)

A standard IBM access method for creating and maintaining data sets at the host. Used by Connect:Enterprise for the VSAM Batch Files.

VSAM Batch Files

A term used for the group of up to 24 files used by the Connect:Enterprise system for storing and maintaining data. The VSAM Batch Files consist of the VSAM Control File, the VSAM Pointer File, the VSAM Batch Queue Files (up to 20), and the VSAM Log Files (up to 2).

VTAM (Virtual Telecommunications Access Method)

An SNA access method used by Connect:Enterprise to receive and send data to a variety of SNA devices or application programs.

VTAM Application Program

A program, such as Connect:Enterprise, that is defined to VTAM and can establish sessions with SNA devices or other VTAM application programs.

X

Xmit once

Specifies that the batch cannot be extracted and that it can be transmitted only one time. After a successful transmit, the batch is permanently locked.

Index