CA XCOM™ Data Transport® for Windows Server/Professional

Overview Guide

r11.5

CA Technologies Product References

This document references the following CA Technologies products:

- CA ACF2® Security (CA ACF2)
- CA Dynam®/T Tape Management (CA Dynam/T)
- CA Top Secret® Security (CA Top Secret)
- CA XCOM™ Data Transport® (CA XCOM Data Transport)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction

This chapter introduces CA XCOM Data Transport. Read this chapter before installing or configuring CA XCOM Data Transport.

This section contains the following topics:

Product Overview

CA XCOM Data Transport is a family of software products that operates under SNA using LU 6.2, or under TCP/IP, to provide high-speed data transfer between supported systems such as mainframes, midrange, PCs, servers, and workstations. You can send files from their local system to remote systems across an SNA network or using TCP/IP, and retrieve files from those remote systems. The same transfer capabilities are available to the local and remote systems.

CA XCOM Data Transport provides a unified solution for communications over more operating systems than any other software product on the market today. CA XCOM Data Transport also has a solid technology base. By using LU 6.2 or TCP/IP communications protocols, CA XCOM Data Transport uses state-of-the-art technology, protecting your company's investment for years to come.

Audience

This *Overview Guide* is written for those personnel who are responsible for installing, configuring, auditing, and administering the product. They may need to customize the product to meet site-specific requirement after the installation and also manage users.

Because CA XCOM Data Transport allows data centers in various locations worldwide to interact with each other for the purposes of sharing data and automating data and report distribution, your site requires different administrators sharing different responsibilities.

The users for this product mainly fall under the following categories:

- Installer
- Database administrators
- System administrators
- Network administrators

How the Data Transport Process Works

To understand the data transport function in a very simplified and generalized way, consider a scenario. For example, when a local system transfers a file to a remote (partner) systems, following steps are performed:

1. Initiation

   The user submits a batch program, starts the menu (the menu interface) or a customer program written using the XCOM API (application programming interface) to initiate the transfer.

2. Information verification

   CA XCOM Data Transport verifies the information contained in the request. For example:

   - When requesting a send file transfer, CA XCOM Data Transport checks whether the local file exists on the local system.
   - When requesting a receive file transfer, CA XCOM Data Transport checks whether the file exists on the remote system.

3. Information confirmation

   If the information is confirmed, CA XCOM Data Transport starts the file transfer.

4. Completion

   The transfer completes and CA XCOM Data Transport logs the details of the transfer in a log.

Note: The previous example illustrates a general idea of how CA XCOM Data Transport works, please be aware that it is simplified; there are many more steps involved in the process. For more information about how data transport works, see the *CA XCOM Data Transport User Guide* for your platform.

Benefits

CA XCOM Data Transport is a widely used, proven vehicle for moving data between a growing number of dissimilar systems. CA XCOM Data Transport provides security, recovery, scheduling, and administrative facilities.

By using CA XCOM Data Transport, you can realize the following advantages:

- Effectively utilize the existing investment in data processing hardware.
- Reduce costs by replacing multiple information transfer products with a single, easy-to-use package.
- Reduce operations and end-user staff training costs by implementing a centrally controlled, highly automated data transfer solution.
- Provide an environment that supports the development of strategic new applications.
- Increase the flexibility and accessibility of remote systems, allowing your organization to respond quickly and accurately to changing business needs.

Applications You Can Design Using the Product

The following applications are a few examples of how CA XCOM Data Transport can be used to interact with other systems company-wide and worldwide:

- Sharing data
- Automating data and report distribution
- Providing unattended back-up to dissimilar systems
- Controlling and auditing network activities
- Maintaining network security
- Communicating with POS (Point-of-Sale) terminals
- Distributing price information from a main data center to departmental stores
- Distributing application software to remote systems
- Providing distributed processing with the *sendjob* function

The applications listed above are only a few examples. Under most conditions, CA XCOM Data Transport allows file sharing between any two systems within your company.

Applications Using the Transfer Function

The key to the considerable flexibility of CA XCOM Data Transport is its ability to transfer the following:

- Files
- Jobs
- Reports

When these functions are combined, a wide variety of applications are possible.

Types of Transfers

CA XCOM Data Transport performs the following transfers:

Sending Files

With CA XCOM Data Transport, a local system can send a data file to be stored on the remote system in a specified remote file.

Sending Reports

CA XCOM Data Transport can send a report to be printed on a remote system.

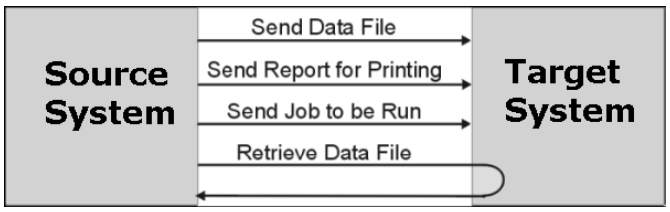
Sending batch jobs for execution

CA XCOM Data Transport can send a job to be executed on a remote system.

Retrieving files

When a system starts the transmission request, it can also retrieve a file from a remote system and store it in a specified local remote file.

The following flow chart illustrates the type of transfers supported by the product:



How Remote Requests are Handled

You can use CA XCOM Data Transport to monitor the network for incoming requests. Upon detecting one, CA XCOM Data Transport determines whether it is a request to send a file inbound (from the remote system to this system) or outbound (from this system to remote system).

File Transfers

You can use the file transfer feature to send or retrieve files from a remote system to a local system.

When CA XCOM Data Transport transfers a file from a remote system to your system, the following actions occur:

- The remote system requests CA XCOM Data Transport to send a file to your system.
- CA XCOM Data Transport allocates memory to the requesting process and opens the file.
- CA XCOM Data Transport then reads the data records from the file.
- CA XCOM Data Transport transfers the file to your system.
- Your system receives the file.

Job Transfers

When CA XCOM Data Transport transfers a job from a remote system to your system, the following actions occur:

- The remote system requests CA XCOM Data Transport to submit a job to your system.
- CA XCOM Data Transport submits the job to your system.
- Your system receives the job file.

Report Transfers

The report transfer feature allows a remote system to send a report to a local system. CA XCOM Data Transport provides a high degree of print redirection and spooling capabilities.

When CA XCOM Data Transport transfers a report from a remote system to your system, the following actions occur:

- The remote system requests CA XCOM Data Transport to send a report to your system
- CA XCOM Data Transport writes the report to an output spool file.
- CA XCOM Data Transport transfers the file to your system.
- Your system retrieves the report from the spool file.

Features

CA XCOM Data Transport provides peer-to-peer communications using LU 6.2 or TCP/IP over a wider range of systems than any other product. All of the major features of CA XCOM Data Transport are supported across the product line.

File Transfer

CA XCOM Data Transport supports high-speed file transfers between all supported operating systems. In some environments, you can start thousands of transfers resulting in hundreds of simultaneous transfers, all with a single operation. Parallel sessions are possible in varying degrees throughout the product line.

You can totally automate CA XCOM Data Transport transfers. On a PC, you can be actively engaged in the use of other applications (for example, word processing) while receiving or transmitting files in the background. Comprehensive management tools allow for effective central-site control of CA XCOM Data Transport activity, including advanced problem determination features.

CA XCOM Data Transport supports transfers between any two systems in an SNA network or a TCP/IP network with one of the following methods:

- By using the z/OS, z/VM, or z/VSE mainframes for store-and-forward
- Through Independent Logical Unit (ILU) support over the SNA (Systems Network Architecture) backbone
- Through Dependent Logical Unit (DLU) support over the SNA (Systems Network Architecture) backbone
- Through use of the TCP/IP network (except for z/VM and Stratus)
- Through use of SSL (Secure Sockets Layer) connections over a TCP/IP network (except for z/VSE, z/VM and Stratus)

PU Type 2.1 Support

CA XCOM Data Transport supports PU Type 2.1 connections to allow the direct interchange of files between the Windows operating environment, NetWare workstations, and others. Support for Independent Logical Units (ILUs) allows CA XCOM Data Transport to deliver data in Advanced Peer to Peer Networking (APPN) and Low Entry Networking (LEN) networks. This means that PCs and midrange that are attached to the same SNA or APPN network can exchange data even if they are not directly connected.

TCP/IP Support

CA XCOM Data Transport provides support for performing transfers using TCP/IP between CA XCOM Data Transport platforms that support TCP/IP and that are running r3.0, r3.1, r11, r11.5 or, r11.6. TCP/IP support is provided between the following platforms:

- i5/OS (AS/400)
- z/Linux
- Linux x86
- NetWare
- Open VMS Alpha
- HP NonStop (Tandem)
- z/VSE
- Windows family
- z/OS
- Most common UNIX platforms

You can use the Secure Socket Layer (SSL) to perform secure TCP/IP transfers between platforms running CA XCOM Data Transport r11 and above that have this support enabled. Users of CA XCOM Data Transport for z/OS have the option of configuring their individual CA XCOM address spaces to use OpenSSL or IBM SystemSSL to encrypt the transmitted data and add a digital signature to the encryption of the transmitted data. Secure TCP/IP support is provided between the following platforms.

- i5/OS (AS/400)
- z/Linux
- Linux x86
- Windows family
- z/OS
- Most common UNIX platforms

Report Distribution

CA XCOM Data Transport allows z/OS, z/VM, z/VSE, i5/OS(AS/400), and OpenVMS Alpha users to take print output from any supported system and automatically transfer it to another system for printing. The application programs producing the reports do not require any modification to support CA XCOM Data Transport report distribution, and no operator intervention is required at either end.

RJE Replacement

Current Remote Job Entry (RJE) systems contain inherent limitations. Remote systems can submit work to the host for processing and receive print data, but the host cannot distribute processing tasks to idle processors residing on the network. A further concern for data processing managers is the requirement that users are familiar with Job Entry Subsystem (JES) commands to operate the system.

CA XCOM Data Transport avoids these limitations by taking advantage of the LU 6.2 and TCP/IP protocols, providing a peer-to-peer relationship between all supported systems. Any CA XCOM Data Transport system is able to send and receive batch jobs and print data from any other CA XCOM Data Transport system without formatting constraints.

For example, an i5/OS (AS/400) user can do the following:

- Automatically retrieve files from a number of attached PCs.
- Process the data.
- Generate a report.
- Send one copy of the report back to the source PC for printing.
- Send another to the z/OS mainframe for printing on a high speed printer.

You can easily implement CA XCOM Data Transport without any changes to your existing application programs. Data is transferred with greater integrity and higher efficiency.

Support of Most Operating Systems

By supporting the LU 6.2 and TCP/IP protocols, CA XCOM Data Transport can transfer data between a diversity of platforms. CA XCOM Data Transport is now available on the following systems:

- HP TRU64 UNIX (Digital UNIX)
- HP Non-Stop (Tandem)
- HP-UX PA RISC
- HP-UX IA64
- IBM AIX
- i5/OS (AS/400)
- z/Linux
- Linux X86
- MS Windows
- NCR 3000 (AT&T)
- Novell NetWare
- Open VMS Alpha
- SCO Open Server
- SCO UnixWare
- Stratus VOS
- Sun Solaris
- z/OS
- z/VM
- z/VSE

Data Link Types

CA XCOM Data Transport supports the following data link types:

- SDLC
- X.25
- Local Area Network (such as Token Ring and Ethernet)
- All SNA data links, including channel-based links
- TCP/IP

Standard Features

The following features are standard to CA XCOM Data Transport:

- Simple installation

  You can install CA XCOM Data Transport without hardware changes to your system.

- Initiation by either system (any-to-any)

  Either system can send and retrieve data files.

- Low maintenance

  There are no hooks or patches into the operating system.

- Choice of interface

  You can choose from batch/command line, programming (on supported platforms), and menu interfaces.

Standard Functions

The following functions are offered over most of the CA XCOM Data Transport platforms:

- Compression

  CA XCOM Data Transport offers a wide range of compression options on most platforms.

- Packing

  CA XCOM Data Transport can pack records into fixed-size data transfer blocks as large as 31K, significantly improving performance and throughput.

- ASCII/EBCDIC translation

  CA XCOM Data Transport can translate data between ASCII and EBCDIC formats as needed. Translations occur on the ASCII-based platform.

- Checkpoint/Restart

  All components of CA XCOM Data Transport support checkpoint/restart. Transfers that are stopped or fail prior to completion automatically resume, continuing from the last checkpoint.

- Store-and-forward

  Users communicating through a common z/OS, z/VM, or z/VSE hub can perform data transfers even if the remote (target) machine is not communicating or turned on at the time of the initial transfer. CA XCOM Data Transport ensures that the data is sent as soon as the device is available.

- Remote spooling

  CA XCOM Data Transport allows z/OS, z/VM, z/VSE, i5/OS (AS/400), and Open VMS Alpha users the following reporting options:

  - CA XCOM Data Transport on all platforms can receive reports.
  - CA XCOM Data Transport on all platforms can send a file to a remote CA XCOM Data Transport partner, requesting that it be treated as a report.
  - Some CA XCOM Data Transport platforms can also take reports off the system spool and forward them to another CA XCOM Data Transport platform without operator action. This automatic report transfer facility is called Process SYSOUT on z/OS and z/VSE, and it is called XQUE on AS/400, HP NonStop (Tandem), and Open VMS Alpha. The z/VM platform does not allow automatic processing of spooled files. However, spooled files on z/VM can be manually received and redirected.

High Capacity and Performance

CA XCOM Data Transport is optimized for high-speed bulk data transfer. For instance, CA XCOM Data Transport for z/OS can allow hundreds of simultaneous file transfers from a single system, depending upon your hardware and software configuration. Comparatively, CICS-based products limit the user to a maximum of 34 simultaneous transfers, and many other VTAM file transfer products are faced with similar limitations.

Security

CA XCOM Data Transport interfaces with PAM or the native security facility on all supported systems that are based on user preference. When security is invoked, you are required to provide a valid user ID and password for the remote system. For example, in the z/OS environment, an interface is also provided to IBM RACF, CA ACF2, and CA Top Secret. Unlike most other communication facilities, CA XCOM Data Transport encrypts passwords. Encrypting passwords ensures that communications line tapping does not breach security.

CA XCOM Data Transport also has special security capabilities that can help data centers handle their individual needs. The security features of CA XCOM Data Transport allow installer specification of what can or cannot run under the privileges of someone other than the person requesting the transmission. These security features can also force user IDs from both remote systems to be the same or different. For otherwise unsatisfied security needs, CA XCOM Data Transport supplies various user exits, which enable user-written security packages to be fully integrated.

CA XCOM Data Transport r11 (on some platforms), r11.5 and r11.6 can also use the Secure Socket Layer (SSL) to perform data transfers under TCP/IP. CA XCOM Data Transport provides certificate authentication, data encryption, and data integrity ensuring all data transfers using SSL are secure.

Management

An important feature for any enterprise-wide information product is the ability to effectively control and manage the distribution of files and work throughout the network. CA XCOM Data Transport systems maintain a comprehensive log of all transfer activity. Utilities are provided to allow the system administrator to view the log online and modify the status of pending or currently active transfers.

Details of any transfer errors are also maintained in the log, allowing rapid problem determination and resolution. In addition, messages signaling the completion of any CA XCOM Data Transport event can be directed to a user in the network.

Chapter 2: Remote System Information

This chapter contains information about important aspects of the operating systems supported by CA XCOM Data Transport that you should be aware of when performing transfers.

For more specific information about operating CA XCOM Data Transport on a specific platform, see the CA XCOM Data Transport guides for that platform and the manufacturer's guides.

The following topics are covered for each platform, as appropriate:

- Naming conventions
- Types of files supported
- Additional features
- Restrictions

This section contains the following topics:

HP NonStop (Tandem)

This section contains information about important aspects of the HP NonStop operating system.

Naming Conventions—HP NonStop (Tandem)

Use the following format to name a HP NonStop file:

```
\<system>.<volume>.<subvolume>.[set the File Name variable]
```

All of these components are restricted to eight characters, except as indicated below.

The following list describes the parts of an HP NonStop file name:

system

Specifies the system name. Up to seven characters.

volume

Specifies the disk name.

subvolume

Specifies a directory name.

filename

Specifies the name of your file.

Example:

The following example uses a volume of $CLX12, a subvolume of SCI, and a file name of FILE1:

```
$CLX12.SCI.FILE1
```

The HP NonStop file system is not a tree structure. Each volume.subvolume is independent, that is, it has no subvolumes above or below.

Types of Files Supported—HP NonStop (Tandem)

CA XCOM Data Transport for HP NonStop supports the following file types through ENSCRIBE, Tandem's disk file architecture:

- Edit files
- Unstructured files

  Unstructured files are large-byte arrays. Data in these files is accessed by using the relative byte address and the READ-COUNT or WRITE-COUNT parameters in the system procedure calls. The application program determines the way in which they are used. An EDIT file is a type of unstructured file signified by the file code 101.

  For more information about ENSCRIBE and unstructured files, see the *ENSCRIBE Programmer's Guide*.
- Structured files

  CA XCOM Data Transport supports entry-sequenced, relative, and key-sequenced structured files:

  - Entry-sequenced files

    Entry-sequenced files are sequential files. Records are stored in the order in which they are entered. These records are variable in length and cannot be added or deleted. They are accessed by their record address.
  - Relative files

    Relative files are ordered by relative record number. The space allocated for each record is specified when the file is created. Records in these files can be deleted and added again in place.
  - Key-sequenced files

    Key-sequenced files are supported only for the Replace operation. The file must already exist for CA XCOM Data Transport to perform an action on it.

File Type Specification—HP NonStop (Tandem)

File type specification differs for send requests and received requests, described as follows:

- Send Requests

  When you send a file from HP NonStop (locally initiated), the remote CA XCOM Data Transport determines the file type when it opens the file.

- Receive Requests

  For locally or remotely initiated receive requests, the file type must be specified by the GUARDIAN_FILE_TYPE parameter. Use one of the following values:

  - EDIT
  - UNSTRUCTURED
  - ENTRY_SEQ
  - RELATIVE

Remotely Initiated Send Requests—HP NonStop (Tandem)

For remotely initiated transfer requests (for example, send a file, job, or report), use the following record formats, which create the indicated Guardian file types:

F

Relative

FB

Entry Sequence

VB

Edit

U

Unstructured

Note: Key sequence files are supported only if the file exists. You can do a replace but not a create.

i5/OS (AS/400)

This section contains information about important aspects of the i5/OS operating system.

Naming Conventions—i5/OS (AS/400)

Use the following format to specify an i5/OS file:

libraryname/filename(membername)

The following list describes the parts of an i5/OS file name:

libraryname

The name of the library that holds the file.

filename

The name of the file you wish to access. Periods are allowed within the file name.

membername

The name of the member in the file. If this component is omitted, it defaults to the file name.

Types of Files Supported—i5/OS (AS/400)

In addition to the standard file type discussed above, the Save File format is also supported. When you wish to send such a file to a System i5 from a z/OS or z/VSE system, the file must exist on the target system prior to your transmission.

Additional Features—i5/OS (AS/400)

XQUE is a CA XCOM Data Transport feature that allows the unattended transfer of reports from output queues to other CA XCOM Data Transport nodes.

XQUE can select specific classes of reports (based on the user, job name, form, and so on) from output queues. XQUE also allows user and workstation groups to be equated to printer destinations on remote CA XCOM Data Transport nodes. You can use XQUE, for example, to get reports back to your host system that are generated on a System i5 that you reach through IBM's HCF facility, or between multiple i5/OS (AS/400) systems connected within a pass-through environment.

Configuration Issues—i5/OS (AS/400)

If you are configuring the VTAM LU that represents the System i5 on a mainframe, make sure that the VTAM USS message 10 is not sent to that LU. IBM's APPC software cannot start a session when this message, commonly called the welcome message, is sent.

To prevent this problem, the VTAM or NCP USSTAB definition must be set to a table that does not have a USSMSG10. The table that IBM originally provided with VTAM is a good alternative because it does not include message 10.

Case Sensitivity—i5/OS (AS/400)

Because the IBM i5/OS is case-sensitive, you must enter the user ID and password in uppercase.

Novell NetWare

This section contains information about important aspects of the Novell NetWare operating system.

Naming Conventions—Novell NetWare

Use the following format to name a Netware file:

Note: CA XCOM Data Transport for LAN Workstation accesses files from any Novell file server in a NetWare network.

[*server*\]*volume:directory**subdirectory*\...*filename*

Types of Files Supported—Novell NetWare

CA XCOM Data Transport for NetWare LAN supports standard NetWare file types.

Destination Printer Information—Novell NetWare

When sending a report to a NetWare system, specify the Destination parameter value or the Destination Printer field in the following form:

*server name**printer queue name*

CA XCOM Data Transport limits the length of this field to 21 characters. The actual name on the destination system can be longer.

Restriction—Novell NetWare

CA XCOM Data Transport for NetWare LAN does not support library transfers to Novell NetWare systems.

OpenVMS

This section contains information about important aspects of the OpenVMS operating system.

Naming Conventions—OpenVMS

Use the following format to name an OpenVMS Alpha file:

device[*directory*]*filename*.*type*;*version*

The entire file specification can be a maximum of 255 characters. The file type can be a maximum of 31 characters.

The following list describes the parts of an OpenVMS file name:

device

Specifies the disk drive name. If the device is not specified, the default provided in the SYSUAF (as defined on the DEC system) for that user is used.

Range: 1 to 15 characters.

Note: The CA XCOM Data Transport remote USERID field determines the SYSUAF USERID.

directory

Specifies the directory and subdirectory information. If this information is not provided, defaults are selected as described under “device” above.

Note: CA XCOM Data Transport accepts angle brackets (< >) in OpenVMS file names, which are converted to square brackets on the DEC system.

Example:

PLAYERS1:<BRIDGES>CARD.DAT
is treated as equivalent to
PLAYERS1:[BRIDGES]CARD.DAT

filename.type

Specifies the specific file within the directory. OpenVMS null file names are used if the file name and type are not provided.

version

Specifies the version of the file. The OpenVMS operating system can keep multiple versions of a file each time that file is saved. It is normal to omit this number to indicate that you want the most recent version of a file, the highest version number.

For more information about OpenVMS file specifications, see the OpenVMS documentation.

Restrictions—OpenVMS

The following restrictions apply to CA XCOM Data Transport for OpenVMS Alpha:

- Specifying transfer type

  All transfers must be TYPE=SCHEDULE (for batch) or QUEUED (from ISPF).

- Non-queued host transfers

  Due to restrictions in the DEC SNA software, the z/OS or z/VSE TYPE=EXECUTE (non-queued) transfer feature fails with an 8003 sense code. It is not supported by CA XCOM Data Transport to an OpenVMS system.

- Operating system

  CA XCOM Data Transport currently supports the OpenVMS Alpha operating system.

- Connectivity

  Specifies the DECNET/SNA software is based on the Physical Unit 2.0 standard and not on the more flexible 2.1. This means that the system must be connected to a VTAM (PU 5) system in an SNA network. CA XCOM Data Transport uses the store-and-forward function (described previously as an additional z/OS, z/VM, and z/VSE feature) to transfer files with other CA XCOM Data Transport partners.

- Multiple session configuration

  Digital does not support parallel sessions with a z/OS or z/VSE system. However, if three file transfers are needed concurrently with an OpenVMS system, it does allow you to define three APPC logical units as a group. A group name can be from one to eight characters. The first character must be alphabetic, while the rest can be any combination of alphanumeric or national characters. Try to use mnemonic names. This feature is useful for assigning nicknames as well.

Example:

The following example calls three logical units, LUD1, LUD2, and LUD3, and assigns them a group name of LAVAX. Code the GROUP and LU parameters for this #PSOTAB entry as follows:

```
GROUP=LAVAX,
LU=(LUD1,LUD2,LUD3)
```

Type LAVAX as the remote system name to schedule transmissions to this VAX through the menu interface. When using the batch interface, use the GROUP parameter instead of the LU parameter. Use the following code:

```
GROUP=LAVAX
```

Groups can be used with all the CA XCOM Data Transport interfaces, including the Process SYSOUT Interface.

- Initiating the session bind request

  Although separate VTAM LU names can be used for CA XCOM Data Transport sessions, you should not LOGAPPL these LUs to CA XCOM Data Transport when configuring on z/OS, z/VSE, or z/VM. This fails with an 0801 or 8003 sense code, because the DEC software must initiate the session bind request.

- Compression options

  Large packing is supported as well as a number of different compression algorithms.

- ASCII-based

  The OpenVMS platform is an ASCII-based system.

Stratus

This section contains information about important aspects of the Stratus operating system.

Naming Conventions—Stratus VOS

Use the following format to name Stratus files:

#top_directory>group_directory>home_directory>filename.suffix

All names must be unique to that level.

Important! CA recommends that you use only UNC conventions for all mapped or redirected drives while sending data to a Windows system.

The following list describes the parts of a Stratus file name:

top_directory

Specifies the physical disks.

Range: 1 to 32 characters.

group_directory

Specifies a group of user home directories.

Range: 1 to 32 characters.

home_directory

Specifies the user's home directory. This directory resides in a group directory.

Range: 1 to 32 characters.

filename

Specifies the name of the Stratus file. Required.

Range: 1 to 32 characters.

suffix

Specifies a file classification. You can have multiple suffixes at the end of a file name. Each suffix starts with a period. The following list describes some common Stratus suffixes for different file types:

source

Suffixes: .pl1, .cobol, .c

Examples: payroll.c, application.cobol

object

Suffixes: .obj

Examples: payroll.obj, application.obj

list

Suffix: .list

Examples: payroll.list, application.list

error

Suffix: .error

Examples: payroll.error, application.error

program module

Suffix: .pm

Examples: payroll.pm, application.pm

command macro

Suffix: .cm

Examples: start_up.cm, compile_and_bind.cm

back up

Suffix: .backup

Examples: payroll.c.backup

Types of Files Supported—Stratus VOS

Stratus supports the following file types for remotely initiated transfers:

- Fixed

  This type of file contains records of the same size. Each record is stored in a disk or tape region holding a number of bytes that is the same for all the records in the file.

- Sequential

  This type of file contains records of varying sizes in a disk or tape region holding approximately the same number of bytes as the record (for example, the record storage regions vary from record to record). Records can only be accessed on a record-by-record basis.

Additional Features—Stratus VOS

The following are additional features of CA XCOM Data Transport for Stratus of which you should be aware:

- Security option

  CA XCOM Data Transport for Stratus can use its own account file to verify the user ID and password and to map the CA XCOM Data Transport user ID to a VOS user ID to check for file access. If this option is turned on and the remote user ID/password combination is invalid, CA XCOM Data Transport for Stratus rejects the request.

- Restart/Recovery facility

  CA XCOM Data Transport for Stratus can attempt periodic data transmissions after the initial file transfer has failed. A certain number of retries can be specified through the xcom_ser.pm file.

Restrictions—Stratus VOS

The following restrictions apply to CA XCOM Data Transport for Stratus:

- No library transfers

  CA XCOM Data Transport for Stratus does not support the transfer of libraries from the mainframe.

UNIX or Linux

This section contains information about important aspects of the UNIX or Linux operating systems.

Naming Conventions—UNIX or Linux

Use the following format to name a UNIX or Linux file:

/directory/subdirectory/.../filename

Use up to 256 characters for the entire path of the file; there are no restrictions on size for the individual parts of the path.

The following list describes the parts of a UNIX or Linux path:

/ (slash)

The root directory when it is in the first position: otherwise, the slash separates directories and file names in the path.

directory

Specifies the directory that contains the file. You can specify more than one directory in a path.

filename

Specifies the name of the UNIX or Linux file.

Types of Files Supported—UNIX or Linux

CA XCOM Data Transport for UNIX or Linux supports standard UNIX or Linux file types.

Restriction—UNIX or Linux

CA XCOM Data Transport does not support library transfers to UNIX or Linux systems.

Trusted Access—UNIX or Linux

CA XCOM Data Transport supports Trusted Access. To use the Trusted Access feature when transferring to UNIX- or Linux-based platforms, note the following:

- The USEROVR and USERPO default table parameters for CA XCOM Data Transport for z/OS or z/VSE must be set to YES.
- Specify USERID=' ', and no passwords in the parameters for the transfer.
- The user ID must be configured for Trusted Access on the UNIX or Linux partner. For more information, see the CA XCOM Data Transport for UNIX and Linux documentation.

Windows

This section contains information about important aspects of the Windows operating systems.

Naming Conventions—Windows

CA XCOM Data Transport supports the standard Windows file names and the Universal Naming Convention (UNC). Some of the file naming conventions are outlined below.

Use the following format to name files when using standard Windows file names:

d:[\][*directory name*\..\]*filename*[.*ext*]

Important! This format may only be used if the drive is a local drive on the Windows system. Do not use for mapped or redirected drives. Use UNC conventions only for mapped or redirected drives.

Note: Use the following format to name files when using UNC file names:

```
\\server name\share name\directory\filename.
```

The following list describes the parts of the file names and UNC file names:

d

Required. Specifies a particular device, indicated as a drive letter. Used for local drives only.

directory name

Required. One or more optional directories and subdirectories.

Subdirectories can take the form of *name[.ext].*

Note: The form of the directory name and file name depend on the operating system running on the server.

filename

Required. Specifies the name of the data file.

For FAT file systems, *filename* is 1 to 8 characters.

NTFS and HPFS file systems support long file names, up to 256 characters, including the extension.

Names may or may not be case sensitive, depending on the file system on the server.

For FAT, NTFS and HPFS, names are not case sensitive. You can use uppercase and lowercase when creating a name, and they display as typed, but internally Windows makes no distinction for this. For example, MYFILE and MyFiLe are considered to be the same file.

Windows also creates an MS-DOS-style name based on the long name for compatibility with environments where long file names are not always supported.

ext

The file extension used to further identify the file.

For FAT file systems, the extension is up to 3 characters.

For NTFS and HPFS, the extension is included in the long file name limit of 256 characters.

Note: If you do not specify an extension, CA XCOM Data Transport does not supply a default.

server name

The name of the server.

share name

The share name is network provider dependent.

For Microsoft Windows networks this is the name of the share.

Types of Files Supported—Windows

CA XCOM Data Transport supports standard Windows file types.

Additional Features—Windows

Additional features include the following:

File Systems

The standard file systems are:

- File Allocation Table format (FAT)
- Windows File System format (NTFS)
- High-performance File System format (HPFS)

File Access

CA XCOM Data Transport accesses files locally or from any file server on the Microsoft Windows Network or the NetWare or Compatible Network, or any other network provider installed on the Windows system.

Security

For transfers to Windows systems running any release of CA XCOM Data Transport:

- Windows is a secured system, because it requires a valid user ID and password (as defined when setting up a user account) to log in to or connect to a server. User IDs and passwords are case sensitive. CA XCOM Data Transport uses the underlying security system to log in to the server as the user defined in the XCOM_USERID parameter. The authority to log on locally is required because there is no facility in CA XCOM Data Transport for Windows 2000, XP, 2003, 2008, or 7 to allow for another domain to be specified.
- When a transfer is sent from another system (such as z/OS or z/VSE), USERID and PASSWORD must be supplied. The following methods model sending from an XCOM z/OS or z/VSE system to a local Windows drive.

Methods of handling Windows systems security from CA XCOM Data Transport:

Employing some security

All users employ the same user ID.

Set XCOM_USERID and XCOM_PASSWORD= to a valid user ID and password that has local logon authority in the xcom.glb file on the Windows side. On the z/OS or z/VSE side, send a transfer with parameter USERID=' ' (blank between two single quotes). This uses the user ID and password from the xcom.glb file.

Employing user level security

Users employ their own user ID and password.

Set XCOM_USERID and XCOM_PASSWORD= to an INVALID user ID and password in the xcom.glb file on the Windows side. On the z/OS or z/VSE side, send a transfer with parameters USERID= and PASSWORD= with a valid Windows user ID that has local logon authority. This causes CA XCOM Data Transport to use the user ID and password supplied. If a password is not supplied, and xcom.glb is checked, the transfer will fail due to the invalid ID and password in the xcom.glb file on the Windows side.

Note: If either of these methods is to be successful, in the case where CA XCOM Data Transport for z/OS or z/VSE is sending to CA XCOM Data Transport for Windows 2000, XP, 2003, 2008, or 7 the CA XCOM Data Transport for z/OS or z/VSE default *tabl*e must have USEROVR=YES. USEROVR=YES is the default. This allows the user ID in the MVS JCL to override the batch job ID. For information about the USEROVR parameter, see the *CA XCOM Data Transport for z/VSE Administrator Guide*.

Trusted Access

CA XCOM Data Transport for Windows Server/Professional supports Trusted Access. To use the Trusted Access feature when transferring to Windows platforms, note the following:

- The USEROVR and USERPRO default table parameters for CA XCOM Data Transport for z/OS or z/VSE must be set to YES.
- Specify USERID=' ' (single quotes without any blanks in between the quotes) and no passwords in the parameters for the transfer.
- The user ID must be configured for Trusted Access on the Windows partner. For more information, see the CA XCOM Data Transport for Windows Server/Professional documentation.

Home Directory

A Windows user can have a default home directory assigned by the Windows administrator.

Destination Printer Information—Windows

When sending a report to a Windows system, specify the Destination parameter value or the Destination Printer field in the following form:

`\\server name\printer queue name`

CA XCOM Data Transport limits the length of this field to 21 characters. The actual name on the destination system can be longer.

Restrictions—Windows

Access to directories and files on drives formatted for NTFS can be controlled with the security features of Windows 2000, XP, 2003, 2008, or 7.

Access to all files on a Windows system can be controlled by the permissions set on a directory or file. The access rights of the user ID on the remote system determine the actions permitted for the transfer. Users cannot use a directory or file unless they have been granted the appropriate permissions.

z/OS

This section contains information about important aspects of the z/OS operating system.

Naming Conventions—z/OS

Use the following format to name a z/OS file (data set):

[*level1.level2.level3...level7*].*level8*[(*membername*)]

The following table describes the parts of a z/OS file name:

level

Specifies the level of a file name. Required.

A file name can consist of multiple levels separated by a period. Each level has the following characteristics:

- It can be up to eight uppercase characters long.
- It starts with either an alphabetic character or a national character ($, #, @, +, -, :, _).

There is a limit of eight levels with a total of 44 characters, including the separating periods.

In most z/OS environments, a data set name is further restricted by security rules created by the installation. Contact the appropriate personnel within your organization for details. Typically, the high-level name (first-level name) must match your z/OS user ID or some other predefined index.

membername

Specifies the particular member in a z/OS partitioned data set (PDS). A PDS is a library containing members that are each separate sequential files. The member name is appended to the end of the file name in parentheses.

Required for z/OS partitioned data sets only.

Range: One to eight alphanumeric or national characters.

Note: Most sites catalog all files through the system master catalog. In short, this means that the system can locate the file you specify by name only. With the rare occurrence of an un-cataloged file, you need to specify the volume and unit information for the device that holds the file.

Example:

The following are examples of valid z/OS data set names:

```
SYS1.VTAMLST
C54684.UTILITY.CNTL(JOBCARD)
PROD.PAYROLL.SEPT90.TIMECARD.DATA
TESTDATA
A.$DDD.LOAD
```

Types of Files Supported—z/OS

Sequential files are the most common forms of data transferred. Individual members of PDS files can also be sent as sequential files. Entire PDS libraries or multiple selected members can be transferred between two z/OS systems or to other systems running CA XCOM Data Transport r11 or higher. PDSE and entire PDSE program libraries are supported in CA XCOM Data Transport starting at r11. PDSE program libraries do not support wildcarding.

All three types of VSAM files (KSDS, ESDS, and RRDS) can be transferred between z/OS systems. These VSAM files must be pre-allocated, or they can be sent to non-z/OS systems as sequential files.

UNIX System Services (USS) files are also supported where an entire file system is stored in a single z/OS data set.

Extended Attribute data sets are supported when running CA XCOM Data Transport r11.6 or higher. This release level also introduces support for Extended Addressability Volumes (EAV).

ISAM, BDAM, IMS, FDR, and DFDSS data sets are not directly supported, but they can be put into a sequential format using native utilities before transmission.

DCB Information—z/OS

The file characteristics for z/OS must be predefined when creating a new file. Collectively, the following characteristics are known as Data Control Block (DCB) parameters:

- Block size
- Logical record length
- Record format
- Volume
- Unit

For more information regarding any of these fields, see the IBM JCL reference manual.

Additional Features—z/OS

You should be aware of the following additional features of CA XCOM Data Transport for z/OS:

- **CICS interface**

  Turn this on by indicating that you want to notify CICS in the appropriate "remote system notify" field in your version of CA XCOM Data Transport. To do this, provide the VTAM APPLID of the CICS system in the related ID field. Your z/OS or CICS application development team can provide you with this. Invoking this interface should start a CICS transaction program following the successful completion of a transfer if one has been provided at the host.

- **Store-and-forward**

  Perform transfers between two nodes connected to an intermediate z/OS system by invoking the indirect transfer feature in your version of CA XCOM Data Transport. This asks you for the final destination LU name and sends a transfer in two stages. The first stage goes to the z/OS JES spool, where it waits for the final destination to be connected before the second stage occurs.

z/VM

This section contains information about important aspects of the z/VM operating system.

Naming Conventions—z/VM

Use the following format to name z/VM files under the CMS operating system:

filename.*filetype*

The two parts can be a maximum of eight characters in length. They can consist of letters, numbers, and/or national characters ($, #, @, +, -, :, _). In general, lowercase letters are not allowed. In the CA XCOM Data Transport for z/VM parameters FILE and LFILE, the file name and file type are specified as one string with a period as a separator.

For minidisk specifications:

- CP OWNER is taken from the volume field, if present. Otherwise, the userid field is used.
- CP address is taken from the unit specification. The default is 191.
- You can have two files with the same file name and file type, but they cannot reside on the same minidisk.

Types of Files Supported—z/VM

The CA XCOM Data Transport Service Virtual Machine runs IBM's Group Control System (GCS) operating system. Due to the limitations of this environment, CA XCOM Data Transport for z/VM only supports the CMS extended file system format. This covers CMS files on minidisks formatted with 512 KB, 1,024 KB, 2,048 KB, and 4,096 KB block sizes.

Note: It does *not* support the following: CMS Shared File System, minidisks formatted with 800-byte blocks, or tape I/O.

DCB Information—z/VM

CMS file characteristics must be predetermined when creating a new file. You must specify the following parameters:

- Record format

  This can be fixed (F) or variable (V).

- Logical record length

  This is the number of characters in the longest line of the file.

Restriction—z/VM

The maximum logical record lengths for different file types are as follows:

- **Disk file**

  32767 bytes

- **Job (RDR file)**

  80 bytes

- **Report (PRT file)**

  133 bytes

z/VSE

This section contains information about important aspects of the z/VSE operating system.

VSAM Naming Conventions—z/VSE

When accessing a file on a z/VSE system, the Remote file name field indicates the file ID as it would be specified on the DLBL (an indicator of whether the file is VSAM or SAM) and, optionally, additional information needed for locating the file.

Format for VSAM File Names

Use the following format to name a VSAM file:

file-id,V[,*catalog-id*]

The following list describes the parts of a VSAM file name:

file-id

Specifies the name given to the data set when it was defined using IDCAMS by including the following line in the JCL:

```
DEFINE CLUSTER (NAME   (file-id)...
```

V

Indicates that this is a VSAM file.

catalog-id

Optional.

The name of the user catalog that owns the VSAM data set as defined using IDCAMS by including the following line in the JCL:

```
DEFINE USERCATALOG (NAME (catalog-id)...
```

Leave this field blank if the data set is owned by the master catalog.

Format for SAM File Names

Use the following format to name a SAM file:

file-id,S,[*unit*],[*location*],[*size*],[*override*]

The following list describes the parts of a SAM file name:

file-id

The name that identifies this data set in the VTOC of the specific DASD volume. This is the file ID you specify on the DLBL JCL statement. Range: 1 to 44 characters.

Note: Do not enclose it in quotes.

S

Indicates that this is a SAM file.

unit

The physical device address as defined by the CUU parameter on the ASSGN JCL statement. It identifies the disk drive on which this file resides. This parameter can be omitted if the UNIT or VOL parameters are specified, or if a DASD manager is in use.

location

Optional for output files.

The starting location of the file on the disk, as defined on the EXTENT JCL statement. If a DASD manager is in use, specify a value of 1.

size

Optional for output files.

Indicates how much space this data set is to use, as defined on the EXTENT JCL statement. For CKD devices, this is the number of tracks. For FBA devices, this is the number of blocks.

override

Optional for output files.

The following override parameters can be specified. The equivalent DFLTAB option is included to describe each override parameter:

- DMYES to force DASDM=YES for this file
- DMNO to force DASDM=NO for this file
- DMEPIC to force DASDM=EPIC for this file

Note: If you are running with a DASD manager, the DASD manager's STRTTRK or Trigger value would be placed in the location field. DASD manager pools should be indicated by putting the pool name in the Volume parameter.

For EPIC/VSE users, you can omit the following:

- The location if you want EPIC to default to its STRTTRK value.
- The size if you want EPIC to default to its DEFEXT value.
- The Volume information if you want EPIC to default to its DEFPOL value.

For CA Dynam/T users who want to access Dynam catalog controlled files (included GDG data sets), no extent information should be entered. (No *cuu*, location, size, or override information and no Volume or Unit parameters for the files you are referencing.)

TAPE Naming Conventions

Use the following format to name a TAPE file:

file-id,T,[*unit*],[*unit*],[*unit*],[*override*]

The following list describes the parts of a TAPE file name:

file-id

Specifies the name that identifies this data set in the tape manager catalog or in the HDR1 label on the tape. This is the file ID you specify on the TLBL JCL statement. Range: 1 to 44 characters.

Note: When the file ID contains imbedded spaces or commas, it should be enclosed in quotes.

Note: IBM only supports a 17-character file ID in a tape header label. If you have a tape manager, 44-character tape file IDs can be supported. CA XCOM Data Transport does not validate your file ID, but takes whatever you put on the statement and passes it to IBM's OPEN routine or to your tape manager as you have entered it.

T

Indicates that this is a TAPE file.

Note: If you enter a transfer request from a platform that has not yet implemented the extended tape processing parameters or menu interface fields for controlling tape processing to a CA XCOM Data Transport z/VSE server, you must use the T option. You are restricted to standard label tape processing.

unit

The physical device address as defined by the CUU parameter on the ASSGN JCL statement. If you are using TAPEM=YES|EPIC, CA XCOM Data Transport ignores any units coded and the tape manager does the tape AVR and assignment. If you are not using the tape manager, the primary assignment is made to the first unit CA XCOM Data Transport finds. Other units found are assigned as temporary alternates.

This parameter can be omitted if you prefer to use the UNIT parameter to specify a unit or two units (primary and alternate). This parameter can be used in conjunction with the UNIT parameter to specify a primary unit and up to four alternate units that are to be assigned by CA XCOM Data Transport prior to open. Units specified on the statement containing the file ID are assigned before units specified on the UNIT parameter. The unit parameter is ignored because tape processing is only supported when you have a tape manager on your z/VSE system.

override

Optional for output files.

The following override parameters can be specified. The equivalent DFLTAB option is included to describe each override parameter:

- TMYES to force TAPEM=YES for this file
- TMNO to force TAPEM=NO for this file
- TMEPIC to force TAPEM=EPIC for this file

Note: The override applies only to the processing for the file whose data set name is on the statement that the override appears on. It is in effect for this transfer only.

VSAM Managed SAM Naming Conventions

Use the following format to name a VSAM managed SAM file:

file-id,M,*prim#recs*, *sec#recs*,*catalog-id*

The following list describes the parts of a VSAM managed SAM file name:

file-id

The name that identifies this data set, which is implicitly defined to VSAM at open time.

Range: 1 to 44 characters.

M

Indicates that this is a VSAM managed SAM file.

Note: If you enter a transfer request from a platform that has not yet implemented the extended tape processing parameters or menu interface fields for controlling tape processing to a CA XCOM Data Transport z/VSE server, you must use the T option. You are restricted to standard label tape processing.

prim#recs

Used for output files only. This indicates the number of blocks (of the size defined by the BLKSIZE parameter) for the primary data set allocation.

sec#recs

Used for output files only. This indicates the number of blocks for the secondary data set allocation. If no secondary allocation is coded, VSAM defaults to 20% of the primary allocation. Zero can be specified if you do not want any secondary allocation.

catalog-id

Optional for output files.

Defines the name of the user catalog that will own the data set. You can leave this field blank if the master catalog owns the data set.

Note: The use of VSAM managed SAM files requires IBM's IDCAMS program to be dynamically loaded in the partition. This requires an additional 130 KB partition GETVIS storage.

DTF Information

z/VSE file characteristics must be predetermined when creating the files. If sending to or receiving from a z/VSE system you must specify the following:

- The record format (RECFM), which can be either fixed (F), fixed blocked (FB), variable (V), or variable blocked (VB).
- The logical record length (LRECL) indicates the number of characters in the longest record in the file.
- The block size (BLKSIZE), which must be one of the following:
  - The LRECL for fixed files
  - A multiple of the LRECL for fixed blocked files
  - The LRECL +4 for variable files
  - The BLKSIZE +4 for variable blocked files

Types of Files Supported—z/VSE

IBM z/VSE supports VSAM (RRDS, KSDS, and ESDS) and SAM files.

Restrictions—z/VSE

The following restrictions apply to CA XCOM Data Transport for z/VSE:

- No FILEOPT=ADD for receiving z/VSE

  CA XCOM Data Transport for z/VSE does not support FILEOPT=ADD if the z/VSE is receiving the file.

- No Checkpoint/Restart for SAM

  CA XCOM Data Transport for z/VSE does not support checkpoint/restart for SAM jobs.

Chapter 3: About SNA Logical Units

This chapter explains the various logical unit (LU) types and discusses independent logical units (ILUs) and other pertinent issues.

This section contains the following topics:

Parts of an SNA Network

This section describes the parts of an SNA network.

LU Connections

An LU is the addressable connection point into an SNA network with which an end-user can send and receive messages. An LU is a set of rules and responsibilities. LUs can be either dependent or independent, and each LU type is associated with a protocol (for example, LU 0, LU 2, LU 3, or LU 6.2). CA XCOM Data Transport only supports LU type 6.2.

The LU provides a connection into SNA for the end-user, which may be either an individual or a transaction program (for example, CA XCOM Data Transport). It allows end-users to communicate with each other and with other network addressable units (NAUs) in the network.

Components of Logical and Physical Network

An SNA network is divided into physical and logical components.

The physical network consists of the following:

- Actual processors called nodes
- Data links between the nodes

The logical network consists of a set of software components called NAUs that include the following:

- Logical units (LUs)
- Physical units (PUs)
- System services control points (SSCPs)

Sessions

A session is a logical connection between two NAUs. Although several types of sessions exist, the end-user is aware of only one type that is LU-to-LU. Sessions are established when one LU sends another LU an SNA request known as a BIND. Each session has its own procedure correlation identifier (PCID).

PCIDs

A PCID is an eight-byte field placed in the BIND, UNBIND, and other SNA requests to help an LU distinguish one session from another. It is required when you are running parallel sessions.

A PCID is also known as a session identifier (SID) in VTAM displays. For each session, VTAM prompts you to note the primary or secondary node and displays the Session ID (SID) in hex. This SID is the PCID. If a trace of the BIND is taken, the PCID vector is towards the end.

The following VTAM operator command lists all sessions generated for that LU:

```
D NET,ID=<luname>,E
```

LUs

This section describes the different LU types.

IBM Strategic LU

LU 6.2 is the only LU type that is crucial to IBM long-term strategy. The 3270 data stream will be moved on top of LU 6.2, and none of the other LU types are strategic (for example, they are not considered in IBM's long-term plans).

LU Types

IBM classifies LUs into roughly seven different types (LU 6.2 is a subset of LU 6). The products that support each of these LU types will continue to be supported in future years and the 3270 data stream will also be preserved, but not in its current form.

The following list shows all of the LU types that are in use today to differing degrees.

0

Denotes a flexible protocol, which eliminates standardization beyond layers of SNA. This was commonly used in the late 1970s (before the advent of LU 6.2).

1

Specifies the protocol used as early as the 1960s by remote job entry (RJE) devices such as the 3770 RJE terminal. Designed for use with printers and card readers, this protocol is most typically used in asymmetrical links where one node is a slave to the host.

2

Specifies the protocol for 3270 video display stations. It defines the data streams used by dumb terminals to communicate with the host.

3

Specifies a variant subset of 3270 protocol that was used to drive printers attached to 3274 cluster controllers. Today it is still used to support old hardware.

4

Specifies a protocol that was intended for use on word processors attached to a host network. You can still see it on old IBM word processors.

6.1

Specifies SNA's prototype protocol defined for program-to-program communication that was developed during the late 1970s. It was a first attempt to provide a standardized mechanism for communication between intelligent peer systems.

6.2

Alternatively referred to by the marketing title Advanced Program-to-Program Communications (APPC), this used to be called the Convergent LU, or the LU type around which the entire IBM product line would converge. LU 6.2 defines standard functions or verbs such as SEND, RECEIVE, and CONFIRM that simplify the work of making two different programs on two different kinds of system talk to each other. This is the only LU type supported by CA XCOM Data Transport.

7

Specifies the data stream of the 5250 video display stations commonly used with the IBM midrange systems.

ILUs

CA XCOM Data Transport supports IUs. An IU is a logical unit that can generate sessions independent of the host. An IU also meets the following criteria:

- It utilizes LU 6.2.
- It works on top of PU 2.1.
- It functions as a primary logical unit and therefore can send a BIND.
- It supports an extended BIND (one that contains a PCID) and works with the Network Control Program (NCP) PU 2.1 support.

Systems that currently support IUs include the following:

- i5/OS(AS/400)
- z/OS
- z/VSE
- MS Windows
- VM (all versions)
- UNIX or Linux
- Netware

LU 6.2 Independent Implementations

Only Type 6 logical units can be independent. All other LU types are dependent. However, not all LU 6.2 implementations are independent.

Not every LU 6.2/PU 2.1 implementation can work with independent LUs. There are some aspects of PU 2.1 that NCP requires with which not all PU 2.1 implementations will work correctly. This reflects the fact that not all midrange and PC SNA Gateway vendors had the latest NCP and VTAM for testing.

PU 2.1 support can be enhanced to work with ILUs without changes to CA XCOM Data Transport. NCP supports ILUs over SDLC, the most common configuration using ILUs. A local area network gateway attached through an SDLC link to a host can also use ILUs. NCP also supports ILUs over a token ring through the TIC.

Direct Sessions with a Dependent Logical Unit

An independent logical unit can have an LU 6.2 session with a dependent LU. This session allows for direct sessions from an i5/OS(AS/400) to an OpenVMS over the SNA background network, even though the VTAM is PU Type 2.0. In this environment, the VTAM LOGAPPL parameter and the VTAM VARY NET LOGON command does not work.

Note: The ILU must initiate the session; it must send to the BIND.

PU Type

When using ILUs with VTAM and NetView displays, VTAM shows the PU type in its status display (PU Type 2 or PU Type 2.1). All PUs originally appears as PU 2.0. Once they become active, they display as PU 2.1.

Index

S

T

U

V

W

Z