

CONNX 14.6 Installation Guide

Table Of Contents

Preface	1
About the Installation Guide	1
Database Terminology	2
Installation Overview	3
CONNX Installation Checklist	4
Upgrade Installation Checklist	5
Displaying Your CONNX Version	6
To display the CONNX version and build numbers	6
How CONNX Works	7
Related Topics	7
How CONNX JDBC Works	8
CONNX JDBC Function Block - UNIX Installation	8
Related Topics	8
Preparing for Installation	10
Step 1 - Installation Checklist and System Requirements	11
General Information	11
Pre-Installation Checklist Procedure	12
To complete the Pre-Installation Checklist	12
System Requirements Checklist	13
Verify System Requirements Procedure	14
To verify system requirements	14
CONNX System Requirements	15
Client and Web Server Requirements	21
UNIX Client System Requirements	22
User Application Requirements	24
CONNX JDBC Client Requirements	26
CONNX JDBC Server Requirements	27
Default Port numbers	28
Step 2 - Install CONNX Administrator and CONNX Client	29
Windows Installation	29
General Information for Installation	29
CONNX Installation Procedure	30
To install CONNX Administrator and CONNX client components	30
UNIX Installation	39
General Information for Installation - UNIX System	39
CONNX Installation Procedure - UNIX	40
CONNX Installation Procedure - Docker	44
Completing CONNX Installation for UNIX ODBC Drivers	48
Completing CONNX Installation for UNIX JDBC Servers	50

CONNX Sample Database Tables	51
Creating CONNX Sample Database Tables	51
To create Oracle sample database tables	52
To build CONNX for DB2 sample tables on the target host via TCP/IP	53
To build CONNX for DB2 sample database tables on the target host via SNA APPC LU 6.2	55
To build Enterprise OLE DB sample database tables	57
To build Enterprise ODBC sample database tables	60
To build Desktop OLE DB sample database tables.....	62
To install Desktop ODBC sample database tables.....	65
To build SQL Server OLE DB sample database tables	67
To build SQL Server ODBC sample database tables.....	70
To build Desktop Informix sample database tables	72
To build Desktop Sybase sample database tables.....	74
To build C-ISAM sample database tables.....	76
Step 3 - Install CONNX JDBC Server and Client.....	78
CONNX JDBC General Information	78
Related Topics	78
CONNX JDBC Server	79
Related Topics	79
Starting and Stopping the CONNX JDBC Server on Windows Systems.....	80
To start the CONNX JDBC Server from the command line on Windows systems.....	80
To stop the CONNX JDBC Server on Windows systems.....	80
Activating the CONNX JDBC Router	81
To activate the CONNX JDBC Router	81
Related Topics	81
Connecting to the CONNX JDBC Server.....	82
To load the JDBC Driver	82
Related Topics	82
CONNX JDBC Configuration Settings	83
The CONNX JDBC Server on Unix	84
Starting and Stopping the CONNX JDBC Server on UNIX.....	84
To start the CONNX JDBC Server on UNIX systems.....	84
To stop the CONNX JDBC Server on UNIX systems.....	85
To verify that the CONNX JDBC Server on UNIX systems is active	85
Using the createJDBCDSN utility script.....	85
To edit the JDBC server startup procedure in UNIX.....	87
Moving a Client to a Non-Windows Platform By Transferring the .jar File.....	89
To move a client to a non-Windows platform.....	89
Related Topics	91
Step 4 - Install CONNX OpenVMS Server.....	92
General Information Regarding the OpenVMS Server	92
Related Topics	92

About the OpenVMS User Account and Password	93
Related Topics	93
CONNX OpenVMS Installation Procedure.....	94
To install the CONNX OpenVMS components on an OpenVMS server	94
Running Multiple Versions of CONNX on an OpenVMS Server	97
Testing the OpenVMS Server Installation	99
Testing User Account Privileges	100
To test the privileges of an OpenVMS user account	100
Using Your Telnet Utility - TCP/IP.....	101
To use Telnet to communicate with the OpenVMS server	101
Step 5 - Install IBM Mainframe-Compatible Server	102
General Information	102
Outline of Installation Procedures	103
z/OS.....	104
Adabas SQL Gateway (CONNX for Adabas) - z/OS	105
Security	106
Prerequisites for Adabas.....	107
Verifying TCP/IP Connectivity.....	107
Installation Procedures.....	108
To install the CONNX components on the target z/OS system.....	108
Introduction	111
To verify installation and start the CONNX Listener process.....	113
To verify installation of CONNX Started Task components.....	113
To enable RACF security	117
To test the CONNX for Adabas user ID/password verification	118
To stop the CONNX for Adabas Batch Job / Started Task TCP/IP Listener	122
SAS/C Network Administration	128
To determine the TCP/IP stack.....	129
Uninstalling the Adabas SQL Gateway (CONNX for Adabas) components	130
Manual Steps: Uninstallation.....	132
Host / Client Security Overview.....	133
IMS.....	139
Prerequisites for IMS - ODBA	139
Verifying TCP/IP Connectivity.....	145
Prerequisites for IMS - DLI.....	146
Verifying TCP/IP Connectivity.....	146
Installing CONNX components on the target IMS system	148
Required and Optional Entry Fields for IMS Server Setup	150
IMS DSN Prefix Data Sets	152
IMS Installation JCL	153
Uninstalling the CONNX IMS components	154
VSAM	155

Prerequisites - CONNX for VSAM / QSAM / PDS	156
Verifying TCP/IP Connectivity.....	156
To install the CONNX components on the target z/OS system	157
Required and Optional Entry Fields for VSAM TCP/IP Server Setup.....	159
DSN Prefix Data Sets.....	161
VSAM Objects for CONNX.STASK.SAMPLES.....	162
Introduction to Started Task installation procedures.....	164
Introduction to installation verification	167
To verify installation of CONNX Started Task components	168
To import metadata from COBOL copybooks.....	173
To import metadata for the CONNX sample QSAM customer physical sequential file	179
To import metadata for the CONNX sample customer partitioned data set	185
To stop the CONNX Started Task/Batch Job TCP/IP Listener	191
To test CONNX for VSAM query/import functionality against SCT VSAM master files.....	192
To test CONNX for VSAM Started Task / Batch Job SCT metadata import functionality	194
To test the CONNX for VSAM Started Task / Batch Job User ID/Password.....	198
Introduction to the CONNX TSO Configuration Utility	200
To stop the CONNX Started Task/Batch Job TCP/IP Listener	201
CONNX TSO Configuration Utility and persistent environment variables	203
To set a persistent CONNX environment variable	204
To change a CONNX persistent environment variable.....	206
To change the number of the CONNX TCP/IP port persistent environment variable.....	207
To convert the persistent CONNX port number to the default.....	209
CONNX Environment Variables: Scope and Precedence	210
To define unique CONNX environment variables per Started Task/Batch Job	211
Environment Variable Security vs. Ease of Use	212
Starting multiple instances of CONNX Started Tasks.....	213
Introduction to uninstallation of CONNX Started Task components.....	218
To uninstall the CONNX for VSAM Started Task components	219
Manual Steps: CONNX for VSAM Started Task Uninstallation	223
Dataset Security	224
CONNX VSAM / QSAM / PDS Host / Client Security Overview.....	224
Default Option - Figure #1:	224
Alternate Option - Figure #2:	224
CONNX Client-Side Security Enhances Host-Side Security	224
Prerequisites - CONNX for CICS/VSAM - z/OS.....	230
Verifying TCP/IP Connectivity.....	230
General Information about the CONNX VSAM Server	231
About the VSAM User Account and Password	232
To install the CONNX components on the target CICS/VSAM z/OS system	233
Required and Optional Entry Fields for VSAM TCP/IP Server Setup.....	235
CONNX DSN Prefix Data Sets.....	237

VSAM Objects for CONNX.CICS.SAMPLES	238
To install the selected components to the OS/390 target	240
Completing CICS Customization	242
To change the CICS startup JCL RPL list.....	243
Installation Verification of CONNX CICS Components	244
To start the IBM CICS TCP/IP sockets interface - IMS	245
To start the CONNX CICS/VSAM TCP/IP Listener	246
To set a CONNX environment variable.....	247
To change a CONNX environment variable.....	248
To convert the CONNX port number to the default.....	249
To display the current CONNX environment variables	250
Running CONNX for CICS/VSAM - z/OS Uninstallation Without Shutting Down CICS	251
VSE.....	252
CONNX for VSAM - VSE	253
Prerequisites - CONNX for VSAM-VSE	253
Verifying TCP/IP Connectivity.....	253
Creating the CONNX Library.....	254
General Information about the CONNX VSAM server	261
To install the CONNX for VSAM-VSE components	262
Required and Optional Entry Fields for VSAM TCP/IP Server Setup.....	264
VSAM Objects for CONNX.CICS.SAMPLES.....	266
To install the selected components to the VSE target	268
For CICS/VSE 2.3 and above installations:	268
Summary of the CONNX for VSAM-VSE Installation Procedure.....	269
ICS/VSE 2.3 External Security Interface: Prerequisites and Setup	272
Completing CICS Customization	274
To define the SAS/C 7.00 CICS extrapartition data queues in the CICS DCT	275
To add SAS/C transient data queue definitions to the CICS Startup JCL	277
Installation Verification of CONNX CICS Components	278
List of Installed Components on the CICS/VSE Partition	280
Configuration Utility	282
To start the CONNX CICS/VSAM TCP/IP Listener	283
To stop the CONNX CICS/VSAM TCP/IP Listener.....	288
To set a CONNX environment variable.....	290
To change the number of the CONNX TCP/IP port.....	291
To change a CONNX environment variable.....	292
To convert the CONNX port number to the default.....	293
To control VSAM file Enables/Opens with CONNX Environment Variables SETENABLE and SETOPEN	294
To enable the Barnard TCP/IP Stack.....	300
To display the current environment variable(s).....	301
Uninstallation Procedures	302

To uninstall CONNX for VSAM - VSE	303
Manual Steps: CONNX for VSE Uninstallation	304
Running CONNX for VSE Uninstallation Without Shutting Down CICS	306
Adabas SQL Gateway (CONNX for Adabas) - VSE	309
Prerequisites for Adabas SQL Gateway - VSE	309
Verifying TCP/IP Connectivity	309
Creating the CONNX Library	310
General Information about the Adabas SQL Gateway Server	317
To install the Adabas SQL Gateway - VSE components	318
Required and Optional Entry Fields for Adabas TCP/IP Server Setup - VSE	320
To install the selected components to the VSE target	322
For CICS/VSE 2.3 and above installations:	322
Summary of the CONNX for Adabas-VSE Installation Procedure	323
Completing CICS Customization - Adabas	325
To add SAS/C transient data queue definitions to the CICS Startup JCL	326
Installation Verification of CONNX CICS Components	327
List of Installed Components on the CICS/VSE Partition	328
Configuration Utility	330
To start the CONNX CICS/Adabas TCP/IP Listener	331
To stop the CONNX CICS/Adabas TCP/IP Listener	332
To set a CONNX environment variable	334
To change the number of the CONNX TCP/IP port	335
To change a CONNX environment variable	336
To convert the CONNX port number to the default	337
To enable the Barnard TCP/IP Stack	338
To display the current environment variable(s)	339
Uninstallation Procedures	340
To uninstall Adabas SQL Gateway - VSE	341
Manual Steps: Adabas SQL Gateway - VSE Uninstallation	342
Running Adabas SQL Gateway - VSE Uninstallation Without Shutting Down CICS	344
Step 6 - Install Adabas SQL Gateway	347
Mainframe Installation of Adabas SQL Gateway	347
Mainframe Installation	347
Unix Installation	348
General Information	348
To install the Adabas SQL Gateway server component on a UNIX machine	349
To build a Docker image for the Adabas SQL Gateway server component on a Linux machine	355
UNIX Environment Variables	360
To stop and restart the Adabas SQL Gateway server component	361
To edit the connxserver shell script	362
To edit the Adabas server startup procedure in UNIX	363
Installation Tips for the Adabas SQL Gateway (CONNX for Adabas)	365

Recommended Configurations for the Adabas SQL Gateway (CONNX for Adabas)	365
Prerequisites - Mainframe	368
Prerequisites - Windows	369
Prerequisites - UNIX	370
Step 7 - Install C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus Server (for Unix only).....	371
General Information for C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus.....	371
To install the CONNX C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus component on a UNIX machine.....	372
To build a Docker image for the C-ISAM and D-ISAM server component on a Linux machine	377
To edit the C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus server startup procedure	382
To stop and restart the C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus server	384
Step 8 - DB2 Host Configuration.....	385
CONNX and DB2.....	385
CONNX DB2 Dynamic SQL Packages	385
To establish CONNX and DB2 CDD configuration options	389
CONNX DB2 Module Connection Scenarios	391
Host Configuration for DB2 Admins	391
CONNX and DB2/UDB (Universal Database) - TCP/IP	392
To establish TCP/IP configuration on DB2 UDB.....	392
To verify server TCP/IP settings via DB2 CLP.....	394
To verify server TCP/IP settings for Linux	395
CONNX and DB2/UDB - APPC SNA LU 6.2.....	396
To verify server APPC/LU 6.2 settings via Control Center	396
DB2/UDB-APPC/LU: To verify the server APPC/LU 6.2 setting via the DB2 UDB Control Center...	396
To verify server APPC/LU 6.2 settings via Command Line Processor	398
To add APPC/LU 6.2 support to the configuration.....	399
CONNX and DB2/UDB - AIX Servers	400
To verify server settings for AIX Servers	400
CONNX and DB2/400 Databases	401
Communications Setup and Scenarios for TCP/IP.....	402
To establish RDBName configuration.....	402
To establish TCP/IP configuration	404
Setting up APPC/LU 6.2 Communications on Target Systems.....	406
To set up MPTN configuration	406
To verify MPTN APPC Controller configuration.....	408
To verify IEEE 802.2 DLC configuration	411
To verify IBMRDB mode description.....	412
Communications Setup and Scenarios for 3-Tier Communications Server	413
To set up the IBM Com Server as an MPTN gateway.....	413
To define the mode name for DRDA connections = IBMRDB	417
To define or verify the DB2 Transaction Program Name	418
To set up the IBM Communications Server in an IEEE 802.2 DLC scenario	423

Communications Setup and Scenarios for MS SNA Server	434
To define the server node	434
To define the server node for a Microsoft SNA Server 4.0	434
Communications Setup and Scenarios for Novell NetWare for SAA 4.0	444
To set up the Novell NetWare in an IEEE 802.2 DLC scenario	444
To configure the node in APPN/APPC for Novell NetWare	445
To configure Data Link Adapters	447
To define a peer link	448
To define a Partner LU 6.2	450
To define a Transaction Program Name	451
To define a mode	452
Communications Setup and Scenarios for 2-Tier SNA Client Configuration	454
To set up 2-tier SNA client configuration for Attachmate - MPTN	454
To set up 2-tier client configuration for Attachmate - IEEE 802.2	459
To set up 2- tier SNA client configuration for RUMBA MPTN	464
To set up 2- tier SNA client configuration for RUMBA IEEE 802.2 DLC	469
CONNX and OS/390 and MVS/ESA Databases	474
Host Configuration	475
To define the RDBName to DB2 DDF	475
DB2/mainframe: To define the RDBName to DB2 DDF	475
To configure DDF for TCP/IP	477
DB2/mainframe: To define the RDBName to DB2 DDF	477
To configure VTAM for DRDA via SNA/LU 6.2	479
Configuring MPTN-specific VTAM Information on the Host	481
To create an SNATCP macro	481
To verify correct IP address and host name	482
DB2 Communications Database Setup	483
Setup	483
APPC Connection Scenarios - 3-Tier Communications Server Configuration	484
To connect to a DB2/MVS or DB2 for OS/390	484
To connect to the target DB2 region via CONNX	490
DB2 Bibliography and Reference Sources	491
References and Sources	491
Step 9 - Post-Installation Checklist	492
General Information Regarding the Post-Installation Checklist	492
Post-Installation Checklist - C-ISAM, DISAM, or Micro Focus	493
Post-Installation Checklist - CONNX JDBC Server	494
Related Topics	494
Post-Installation Checklist - CONNX Upgrades	495
Post-Installation Checklist - DataFlex	496
Related Topic	496
Modifying the CONNX Registry File Settings for DataFlex Year Digits	497

To modify the CONNX.INI file for DataFlex Year Digits	497
Related Topic.....	497
Post-Installation Checklist - DB2.....	498
Related Topic.....	498
Post-Installation Checklist - IBM Mainframe-compatible Server.....	499
Post-Installation Checklist - Adabas SQL Gateway (CONNX for Adabas).....	500
Post-Installation Checklist - Informix data sources	501
Post-Installation Checklist - Oracle	502
Related Topic.....	503
Post-Installation Checklist - POWERflex	504
Post-Installation Checklist – OLE DB- and ODBC-Compliant Data Sources	505
Post-Installation Checklist - OpenVMS Server	506
Related Topic.....	507
Editing the OpenVMS Site-Specific Startup Command	508
To edit the OpenVMS Site-Specific Startup Command Procedure	508
Post-Installation Checklist - Sybase data sources	510
Troubleshooting Post-Installation Procedures	511
Testing a Database Connection.....	511
To test the database connection using InfoNaut	511
Reviewing the Minimum OpenVMS Quotas Procedure.....	515
To review the minimum recommended quotas.....	515
Minimum Quota Values for RMS, Rdb, and DBMS Databases	517
Related Topics	517
Converting Data Dictionaries	518
To convert CONNX Data Dictionaries to the latest version	518
Troubleshooting Connection Problems.....	520
Related Topics	520
Troubleshooting with RMS, Oracle Rdb, and Codasyl DBMS Databases.....	521
Related Topics	521
Troubleshooting with TCP/IP and OpenVMS.....	522
Related Topics	522
Pinging the OpenVMS Server - TCP/IP	523
To verify the TCP/IP network connection with the OpenVMS server	523
Related Topic.....	523
Reviewing CNXSETUP.LOG Using FTP - TCP/IP	524
To copy CNXSETUP.LOG using FTP	524
Related Topics	525
CONNX JDBC Sample Application	526
CONNX JDBC Sample Application.....	526
Related Topics	526
Using the CONNX JDBC Sample Application in a Microsoft Windows Environment.....	527
To start the CONNX JDBC Sample Application in a Windows environment.....	527

Related Topic.....	527
Using the CONNX JDBC Sample Application in a non-Windows Environment	528
To start the CONNX JDBC Sample Application in a non-Windows environment.....	528
Running the CONNX JDBC Sample Application	529
To connect to the data source	529
Related Topic.....	530
To choose an Output Type.....	531
To choose a Query Type.....	532
To create a one-parameter query	533
To create a two-parameter query.....	534
Navigator Bar	535
CONNX JDBC Sample Applet.....	536
Running the CONNX JDBC Sample Applet.....	536
To run the CONNX JDBC Applet	537
Step 10 - License Registration	540
General Information Regarding License Activation	540
License Activation	541
Updating an existing license	543
Notes about License Servers and Licenses	545
Offline License Activation.....	547
Offline License Revoke	551
Moving Licenses from one License Server to Another	554
Emergency License Activation	555
Frequently Asked Questions about Licenses.....	559
Upgrading and running multiple versions of CONNX	560
Backward and forward compatibility	560
The CONNX Client	560
The JDBC Jar File	560
Using multiple version of CONNX simultaneously	561
The CONNX Client	561
The CONNX Data Server & Listener.....	561
Silent CONNX Client Installation.....	563
Introduction to Silent Client Installation.....	563
Related Topics	563
Creating the CONNX Client Installation Record File Procedure.....	564
To create the CONNX client installation record file	564
To Create a Shortcut to CONNX Setup	571
Related Topics	571
Running the Silent CONNX Client Installation Procedure	572
To run a silent CONNX client installation.....	572
Related Topic.....	573
Glossary	574

Terms and Abbreviations574

Index.....577

Preface

About the Installation Guide

This installation guide guides you through a step-by-step process for installing, configuring, and testing CONNX in Microsoft® Windows®, UNIX-, and Java-compatible operating system environments. This help file is to be used with the CONNX User Reference Guide, also available online on the Software AG documentation web site.

Database Terminology

For consistency, database references in this manual indicate the following:

Databases marked with a single asterisk (*) are supported from the Unix client via the CONNX Enterprise Server Service (see the CONNX User Reference Guide for procedural instructions). Databases marked with a double asterisk (**) include support for both Windows and Unix databases. The databases on Windows are accessed through the CONNX Enterprise Server Service from the Unix client.

RMS** – Compaq Record Management Services (RMS), a component of the VMS/OpenVMS operating system.

Rdb** – Oracle Rdb product, a relational database product originally developed by Digital Equipment Corporation.

DBMS** – The CODASYL DBMS product, a non-relational database product originally developed by Digital Equipment Corporation.

DataFlex* – Data Access Corporation products.

POWERflex* – POWERflex Corporation products.

DB2* – IBM DB2 products.

Oracle* – Oracle 8i and above.

OLE DB and ODBC Providers* – Any OLE DB or ODBC Provider-compliant data source. Examples include Sybase and Microsoft SQL Server, among others.

JDBC** – Java Database Connectivity.

VSAM** – Virtual Storage Access Method, which supports key- or entry- sequenced and relative-record file systems on IBM mainframe operating systems (MVS, OS/390, VSE, z/OS).

IMS - Information Management System. Any of several system environments available with a database manager and transaction processing, capable of managing complex databases and terminal networks.

C-ISAM** – Indexed Sequential Files to which CONNX provides read/write access. With CONNX, C-ISAM files can resemble a relational database management system. Available on both Unix and Windows platforms.

DISAM** – With CONNX, DISAM files can resemble a relational database management system. Available on both Unix and Windows platforms.

Micro Focus** – Micro Focus product produced by Micro Focus International Limited. With CONNX, Micro Focus files can resemble a relational database management system. Available on both UNIX and Windows platforms.

PostgreSQL – PostgreSQL product produced by the PostgreSQL Global Development Group and the University of California.

Adabas - Adabas product produced by Software AG, and is available from the HP-UX**, Linux**, AIX**, Solaris**, and Windows* clients. The CONNX for Adabas module is also known as the Adabas SQL Gateway.

Installation Overview

There are ten steps to complete for a full installation of CONNX. It is not necessary to complete all steps in order to use CONNX.

Step 1 - Installation Checklist and System Requirements

This step helps verify the [information](#), [media](#), hardware, and [software](#) needed to install CONNX.

Step 2 - Install CONNX Administrator

This step installs the CONNX [administrator](#) and client software on a computer.

Step 3 - Install CONNX JDBC Server and Client

This step installs the [CONNX JDBC](#) server and client software on a computer.

Step 4 - Install CONNX OpenVMS Server

This step installs the [CONNX server](#) software on your OpenVMS system. The server software is required when using Rdb, DBMS, and RMS databases. This step is not necessary for all other databases.

Step 5 - Install IBM Mainframe-compatible Server

This step installs the CONNX server software on your z/OS or VSE server. The server software is required when using VSAM or IMS files. This step is not necessary for all other databases.

Step 6 - Install Adabas Server (IBM mainframe only)

This step installs the Adabas SQL Gateway server component, also known as CONNX for Adabas, on your z/OS mainframe. It is not required for Windows or UNIX operating system installations.

Step 7 - Install C-ISAM, DISAM, or Micro Focus Server (for UNIX only)

This step installs the CONNX C-ISAM, DISAM, or Micro Focus server software on your selected system. This step is not necessary for all other databases.

Step 8 - DB2 Host Configuration

This step describes host configuration for IBM mainframe-compatible servers.

Step 9 - Post-Installation Checklist

This step assists in configuration of the CONNX client, administrator, and server software.

Step 10 - License Activation

This step activates the CONNX license(s) and enables the product for use.

CONNX Installation Checklist

The table below contains the CONNX Installation Checklist. You can print a copy of this for your records. As each step is completed, place a check mark in the appropriate column, and then continue to the next step.

CONNX Installation Checklist

Check here when item is complete	Step
	Completed Step 1 - Installation Checklist and System Requirements
	Completed Step 2 - Install CONNX Administrator and CONNX client
	Completed Step 3 - Install CONNX JDBC server and client
	Completed Step 4 - Install CONNX OpenVMS server [Rdb, DBMS, and RMS only]
	Completed Step 5 - Install IBM Mainframe-compatible Server
	Completed Step 6 - Install Adabas Server (for IBM mainframe and UNIX only)
	Completed Step 7 - Install C-ISAM, DISAM, or Micro Focus Server (for UNIX only)
	Completed Step 8 - DB2 Host Configuration
	Completed Step 9 - Post-Installation Checklist
	Completed Step 10 - License Registration

Upgrade Installation Checklist

The table below contains the Upgrade CONNX Installation Checklist. You can print a copy of this for your records. As each step is completed, place a check mark in the appropriate column, and then continue to the next step. Use the Upgrade CONNX Installation Checklist for the following:

- **Increasing the number of licensed connections for CONNX**
When upgrading from a 25-user license to a 50-user license, for example
- **Adding CONNX products**
When adding CONNX for Oracle to a CONNX installation that only includes CONNX for RMS, for example
- **Version Upgrade**
When upgrading from CONNX 8.0 to 9.0, for example
- **Service Pack**
When installing a Service Pack that upgrades CONNX from build 1 to build 2, for example

Upgrade CONNX Installation Checklist

Adding Users	Adding Features	Version Upgrades	Service Packs	Step
	X	X		Completed Step 1 - Installation Checklist and System Requirements
	X	X	X	Completed Step 2 - Install CONNX Administrator and CONNX Client
				Completed Step 3 - Install CONNX JDBC Server and Client
	X	X	X	Completed Step 4 - Install CONNX OpenVMS Server (Rdb, DBMS, and RMS only)
	X	X	X	Completed Step 5 - Install IBM Mainframe-compatible Server
	X	X	X	Completed Step 6 - Install Adabas Server (for IBM mainframe only)
	X	X	X	Completed Step 7 - Install C-ISAM, DISAM, or Micro Focus Server (for UNIX only)
	X	X		Completed Step 8 - DB2 Host Configuration
X	X	X		Completed Step 9 - Post-Installation Checklist
X	X	X*		Completed Step 10 - License Activation

* License Activation required only when upgrading from a non license server version of CONNX

Related Topics

 [Displaying Your CONNX Version](#)

 [License Activation](#)

Displaying Your CONNX Version

When performing a version upgrade, review exactly which CONNX version and build are currently installed. Complete the following procedure to display the current CONNX version and build numbers.

To display the CONNX version and build numbers

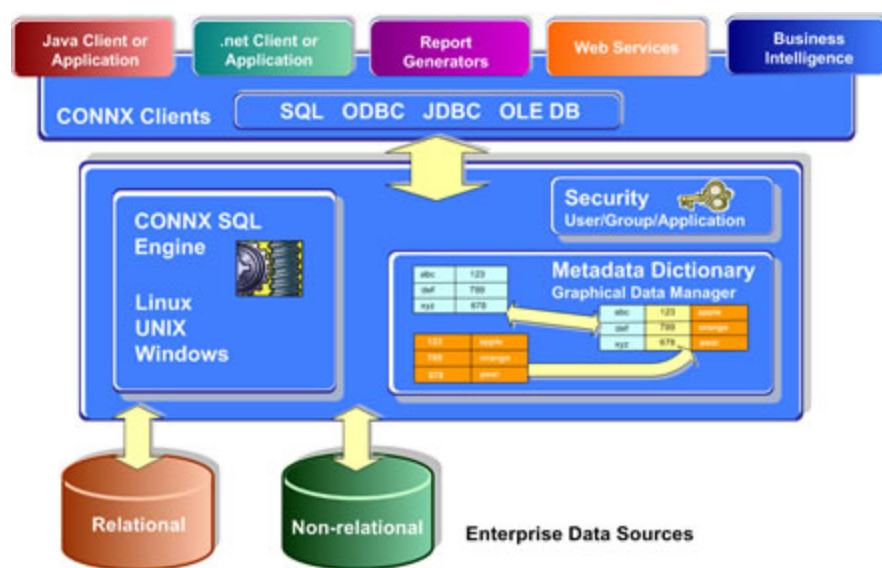
1. From the Windows File Explorer select **C:\Windows**, and then click the **System** folder.
2. In the **System** (or **System32**) folder, right-click the **CONNX32.dll** file, and then select **Properties**.
3. Select the **Version** tab, and then click **Product Version** in the **Item Name** list box. The CONNX version and build numbers are displayed.

How CONNX Works

The diagram, CONNX Function Block, shown below, is an overview of CONNX installation and operation. The CONNX Administrator and CONNX client components are installed on a selected computer, usually designated as the CONNX Administrator computer, although it can be installed on more than one computer. Installation of the CONNX client provides an interface between the end-user application and databases of several different types and platforms. To the end user, however, the databases, tables, joins, views, and other database elements appear as if all the various data sources were a single database.

- CONNX also installs a server component for the RMS, Rdb, and DBMS databases on an OpenVMS server.
- DB2, Oracle, SQL Server, PostGres, MySQL, Sybase, Informix, OLE DB Enterprise, OLE DB Desktop, DataFlex, and POWERflex databases do not require a CONNX server component.
- The relational data sources Oracle Rdb, Oracle, SQL Server, Informix, Sybase, Adabas, and PostGreSQL are supported on Windows and Unix systems.
- VSAM, Adabas, and IMS databases require an IBM mainframe-compatible server.
- Adabas is available in mainframe, Windows, and UNIX versions.
- C-ISAM, DISAM, and Micro Focus install a server component on the UNIX server. There are also Windows-compatible versions.
- The CONNX OLE DB and ODBC server component resides on the client machine. It uses the OLE DB or ODBC driver that exists on the client machine. The CONNX OLE DB server component is automatically installed when required.
- OLE DB is not supported when CONNX is run on a non-Windows client machine. In such cases, a third-party ODBC driver manager can be used.

CONNX Function Block



Related Topics

- >> [CONNX System Requirements](#)
- >> [Client and Web Server Requirements](#)
- >> [CONNX Installation Procedure](#)
- >> [CONNX OpenVMS Installation Procedure](#)

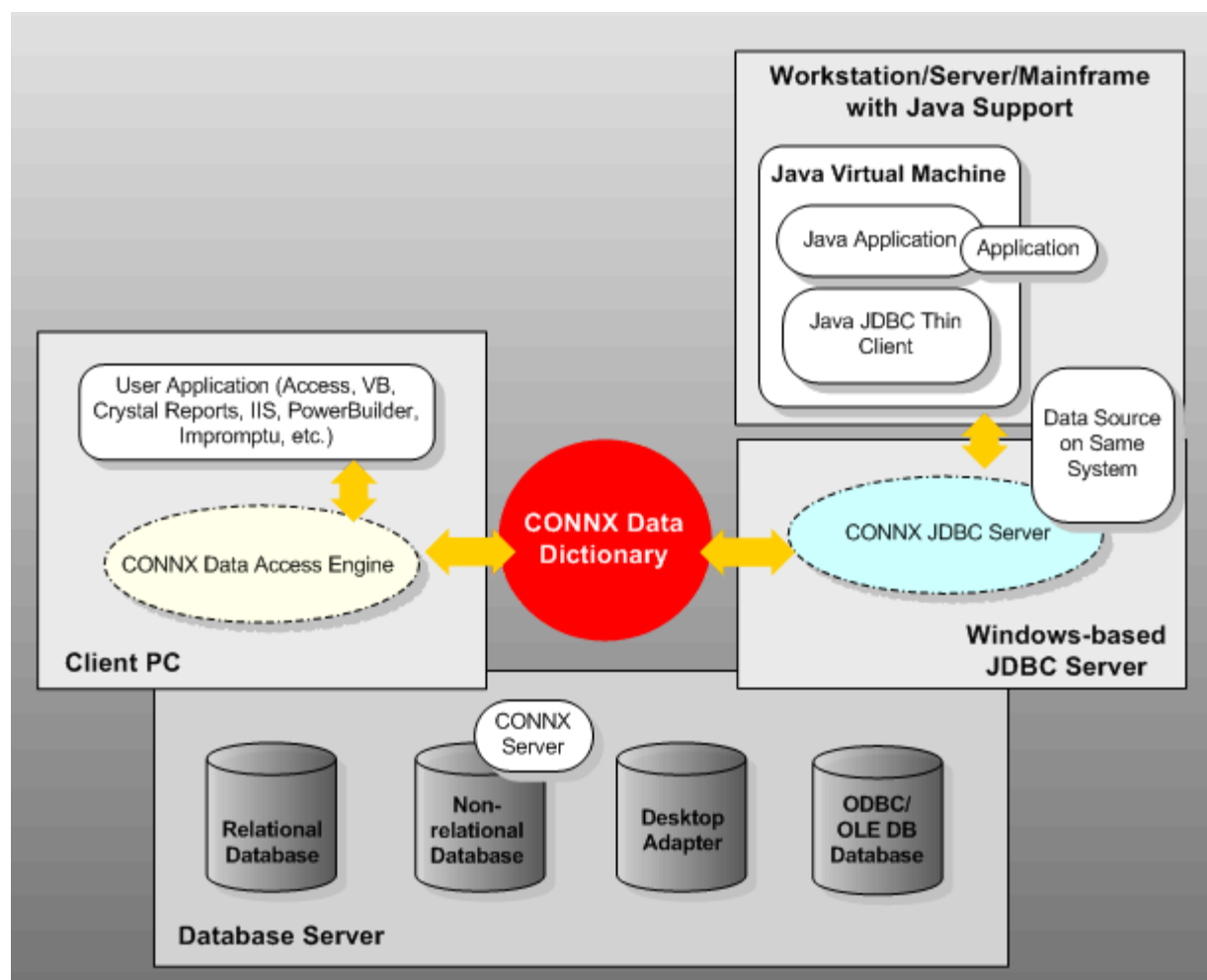
How CONNX JDBC Works

The diagram, CONNX JDBC Function Block, shown below, is an overview of CONNX JDBC installation and operation. The CONNX Administrator and CONNX client components are installed on a selected computer, usually designated as the CONNX Administrator computer, although it can be installed on more than one computer. Installation of the JDBC Thin Client provides an interface between the end-user application and databases of several different types and platforms. To the end user, however, the databases, tables, joins, views, and other database elements appear as if all the various data sources were a single database.

The CONNX JDBC Server component is responsible for handling JDBC requests from the JDBC Thin Client and must reside on a shared server.

Diagrams displaying the both the CONNX JDBC Function Block and the CONNX JDBC Function Block for UNIX installations are below.

CONNX JDBC Function Block



CONNX JDBC Function Block - UNIX Installation

Related Topics

[CONNX JDBC Client Requirements](#)

 [CONNX JDBC Server Requirements](#)

Preparing for Installation

Do not install a CONNX product unless a valid license has been purchased or an evaluation license has been issued for the product. The Software AG End User License Agreement (EULA) governs the use of CONNX products.

Software must not be transferred from one computer to another or to another portion of a network before CONNX registration procedures have been completed or without the written consent of CONNX Solutions.

This guide assumes that the user has some familiarity with the OpenVMS Operating System, or that of the server upon which your database resides, and with the operating system on the PC on which the CONNX client software is installed. CONNX client software is available for Microsoft®Windows operating systems, Linux, HP-UX, Solaris, AIX, and Java-compatible platforms. For information about the OpenVMS operating system, consult OpenVMS documentation or Compaq technical support. For information on a Windows computer's operating system consult the accompanying documentation, the online help for the operating system, or Microsoft technical support.

Step 1 - Installation Checklist and System Requirements

General Information

Prior to starting CONNX installation, print out and complete the Pre-Installation Checklist Procedure. Check off each item in the following table as they are completed:

Pre-Installation Checklist

Check here when item is complete	Item
	1. Have the CONNX installation executable
	2. Have the CONNX License Files for each database platform
	3. Have an ODBC or OLE DB driver installed if accessing an OLE DB- or ODBC-compliant data source
	4. Have FTP enabled on both the install PC client and the target host

Pre-Installation Checklist Procedure

The Pre-Installation Checklist can be completed with the following procedure.

To complete the Pre-Installation Checklist

1. Confirm that you have the CONNX installation executable.
2. Confirm that you have the CONNX License files for each database platform.
3. If accessing an OLE DB- or ODBC-compliant data source, confirm that an OLE DB or ODBC driver is installed.
4. Confirm that FTP is enabled on both the install PC client and the target host (Mainframe or Linux/Unix).

System Requirements Checklist

Print out and complete the Verify System Requirements procedure and check off each item in the System Requirements Checklist below. This procedure verifies that your system components and configuration meet CONNX requirements. Requirements are included for CONNX in both a 2-tier enterprise solution or 3-tier enterprise solution.

System Requirements Checklist

Check here when item is complete.	Item
	1. Meets system requirements (see CONNX System Requirements).
	2. Meets client and Web server requirements (see Client and Web Server Requirements).
	3. Meets Java client and server requirements (see CONNX JDBC Client Requirements and CONNX JDBC Server Requirements).
	4. Meets user application requirements (see User Application Requirements).

Verify System Requirements Procedure

Use the following procedure to verify system requirements.

To verify system requirements

1. Confirm that your system meets the system requirements listed in the [CONNX System Requirements](#)). When confirmed, check Item 1 in the [System Requirements Checklist](#).
2. Confirm that your Microsoft Windows system and/or Linux/nix system meets the client and Web server (if used) requirements listed in [Client and Web Server Requirements](#). When confirmed, check Item 2 in the [System Requirements Checklist](#).
3. Confirm that your system meets the Java client and server requirements listed in [CONNX JDBC Client Requirements](#) and [CONNX JDBC Server Requirements](#). When confirmed, check Item 3 in the [System Requirements Checklist](#).
4. Confirm that your system meets the user application requirements listed in [User Application Requirements](#). If your application or programming language is not listed in the table, contact your CONNX representative for more information. When confirmed, check Item 4 in the [CONNX System Requirements Checklist](#).
5. Unix/Linux only: CONNX uses the GNU C library and requires the glibc runtime. Confirm that your system has the glibc package installed. On a 64bit system CONNX uses both the 32-bit and 64-bit glibc packages so both need to be installed. On 64-bit Linux systems, the 32-bit compatibility packages must be installed.

This completes the Verify System Requirements Procedure. Check the Step 1 box on the [CONNX Installation Checklist](#), and then continue to Step 2 - Install CONNX Administrator and CONNX Client.

CONNX System Requirements

The following table lists CONNX system requirements, including hardware requirements, for applicable databases and their operating systems.

CONNX Client	
Windows	
Hardware	N/A
Network	TCP/IP
Operating System	Windows Server 2012/Windows 10 and above
Memory	16 GB
Disk Space	40 GB
IBM Mainframe	
Hardware	FTP or SFTP server required for installation only. VSE also requires CICS-TS.
Network	VSE only: CSI TCP/IP (OE stack); Barnard (Build 257 pre32 or newer) TCP/IP connectivity stack available
Operating System	z/OS and zVSE
Memory	
Disk Space	
Security	OSS segment must be defined for the user ID in the security system (RACF / ACF2 / Top Secret) or it must be available by default.
HP-UX	
Hardware	Processor: PA-RISC and Itanium, 512 MB Disk space: Installing the optimized version of Adabas requires approximately 45MB. An additional 30 MB are required if a demo database is created during installation. An additional 60 MB are required if you install the trace version of Adabas. These figures do not include disk space requirements for other databases that you will create.
Software	HP-UX 11.0 (64-bit) or HP-UX V11.11i (64-bit) or HP-UX 11.11 (32-bit)
Solaris	
Hardware	Processor: UltraSPARC Memory: 512 MB Disk space: Installing the optimized version of Adabas requires approximately 45MB. An additional 30 MB are required if a demo database is created during installation. An additional 60 MB are required if you install the trace version of Adabas. These figures do not include disk space requirements for other databases that you will create.
Software	Operating System: Solaris Version 8 or above

AIX	
Hardware	<p>Processor: IBM e-Server P-Series or RS/6000</p> <p>Memory: 512 MB</p> <p>Disk space: Installing the optimized version of Adabas requires approximately 45MB. An additional 30 MB are required if a demo database is created during installation. An additional 60 MB are required if you install the trace version of Adabas. These figures do not include disk space requirements for other databases that you will create.</p>
Software	Operating System: IBM AIX5L Version 5.1 system maintenance level 2 (64-bit) or Version 5.2
Linux	
Hardware	<p>Processor: Intel, IBM Mainframe</p> <p>Memory: 4 GB</p> <p>Disk space: Installing the optimized version of Adabas requires approximately 45MB. An additional 30 MB are required if a demo database is created during installation. These figures do not include disk space requirements for other databases that you will create.</p>
Software	Operating System: Linux kernel 2.6.18 or above for Intel and zLinux platforms

RMS (any version)	
Hardware	Compaq VAX Server, Compaq AlphaServer, Itanium
Network	TCP/IP
Operating System	OpenVMS
Memory	12 MB VAX, 32 MB Alpha
Disk Space	20,000 blocks available (10 MB)
Codasyl DBMS	
Hardware	Compaq VAX Server, Compaq Alpha Server, Itanium
Network	TCP/IP
Operating System	OpenVMS
Memory	12 MB VAX, 32 MB Alpha
Disk Space	20,000 blocks available (10 MB)
Oracle Rdb	
Hardware	Compaq VAX Server, Compaq Alpha Server, Itanium

Step 1 - Installation Checklist and System Requirements

Network	TCP/IP
Operating System	OpenVMS
Memory	12 MB VAX, 32 MB Alpha
Disk Space	20,000 blocks available (10 MB)
Oracle version 8i and above	
Hardware	Compaq VMS Server (VAX, Alpha or Itanium), Personal Computer (Intel)/Alpha Sun/UNIX Workstation
Network	TCP/IP
Software	CONNX Requires the Oracle Client Interface Software (OCI) to be installed. This can be the standard Oracle Client, or the Oracle Instant Client. OCI supports Bulk operations only when the version of the Oracle Client matches the version of the Oracle database. This is an important performance consideration with DataSync and Event Replication. For best performance, ensure that the version of the Oracle client is the same as the Oracle database being accessed. On Linux/Unix, CONNX contains an embedded OCI, and no additional software is required.
Operating System	OpenVMS (any version), Windows Server 2012 and above, UNIX/Linux (any version)
Memory	N/A
Disk Space	N/A
DataFlex version 2.3 and above	
Hardware	Personal Computer (Intel)/Alpha, UNIX Workstation
Network	Any file-sharing protocol
Operating System	Windows Server 2012 and above, Linux
Memory	N/A
Disk Space	40MB (Linux)
POWERflex	
Hardware	Personal Computer (Intel)/Alpha, UNIX Workstation
Network	Any file-sharing protocol
Operating	Windows Server 2012 and above, Linux

System	
Memory	N/A
Disk Space	40MB (Linux)
OLE DB Providers (Sybase Informix, Microsoft SQL Server, Microsoft Access)	
Operating System	Windows Server 2012/Windows 10 and above
ODBC Providers (Must support ADO and be fully ODBC Level 2 compliant.)	
Operating System	Windows Server 2012/Windows10 and above
C-ISAM	
Hardware	HPUX, Solaris, AIX, Linux, Personal Computer (Intel)
Network	TCP/IP
Operating System	HPUX 10.2+, HPUX64, SUN OS 5.6, SUN OS 5.7+, AIX 4.3+, AIX 4.2, Linux kernel 2.6.18 or above. Windows Server 2012/Windows 10 and above.
Memory	N/A
Disk Space	40 MB
DISAM	
Hardware	HPUX, Solaris, AIX, Linux, Personal Computer (Intel)
Network	TCP/IP
Operating System	HPUX 10.2+, HPUX64, SCO Server, SUN OS 5.7+, AIX 4.3+, AIX 4.2 or Linux kernel 2.6.18 or above Windows Server 2012/Windows 10 and above
Memory	N/A
Disk Space	40 MB

Step 1 - Installation Checklist and System Requirements

Micro Focus	
Hardware	HPUX, Solaris, AIX, Linux(Intel), Personal Computer (Intel)
Network	TCP/IP
Operating System	HPUX 10.2+, AIX 4.3+, AIX 4.2, Linux kernel 2.6.18 or above, Solaris 5.8 or 5.9 with Micro Focus Server Express 4.0 SP1 only Windows Server 2012/Windows 10 and above
Memory	N/A
Disk Space	40 MB
Adabas	
Hardware	HPUX, Solaris, AIX, Linux(Intel and zLinux), z/OS, VSE, Personal Computer (Intel)
Network	TCP/IP
Operating System	HPUX 10.2+, AIX 4.3+, AIX 4.2, Linux, Linux/390, Solaris 5.8 or 5.9 z/OS VSE 2.4 and above Windows Server 2012/Windows 10 and above
Memory	N/A
Disk Space	40 MB

DB2 Product	Operating System
DB2/6000; DB2 UDB for AIX	AIX 4.3 and above
DB2 UDB for z/OS and OS/390	z/OS
DB2/400 V3R1 and above	OS/400
DB2/400 V4R2 and above; DB2 UDB for iSeries	OS/400 and iSeries
DB2 UDB Enterprise Server Edition	Windows Server 2012/Windows 10 and above
DB2 UDB for	Linux

Linux Enterprise Server Edition	
--	--

VSAM Product	Operating System	Supported File Types	Network Software	CICS Version/Release
CONNX for CICS/VSAM	z/OS	VSAM	TCP/IP V3R2 and above	V4R1 or TX 1.x and above
CONNX for VSAM / QSAM / PDS	z/OS	VSAM / QSAM / PDS	TCP/IP V3R2 and above	N/A
CONNX for CICS/VSAM	VSE 2.3 and below	VSAM	TCP/IP (CSI / IBM) or Barnard TCP/IP stack	V2R3 and below
CONNX for CICS/VSAM	VSE 2.4 and above	VSAM	TCP/IP (CSI / IBM) or Barnard TCP/IP stack	TS 1.1.1 and above

IMS Product	Operating System	Supported File Types	Network Software	CICS Version/Release
CONNX for IMS	z/OS	IMS	TCP/IP V3R2 and above	N/A

CONNX has been tested or certified with the following TCP/IP software products on OpenVMS: UCX 3.0 and above, Multinet and TCPware.

OLE DB/ODBC providers use third-party data providers which have their own hardware and network requirements and are installed on the client machine.

The server requirements are dictated by the third-party driver selected for operation.

Client and Web Server Requirements

Requirement	Minimum	Recommended
Processor	Intel Core or compatible processor	Intel core or compatible processor
Memory	8 GB	32 GB
Free Disk Space	25 GB	50 GB
Operating System and Server	Windows Server 2012/Windows 10 and above Microsoft® IIS® 8 or above	Windows Server 2012/Windows 10 and above IIS 8 or above
Software (Three-tier Enterprise)	Any web browser that supports HTML 3 or later	
Network Connectivity	TCP/IP (Winsock 1.1-compliant)	Microsoft TCP/IP (Winsock 2.9-compliant)
Software OLE DB- or ODBC-compliant	OLE DB or ODBC driver and established file, system, or user DSN on all clients (file DSNs are recommended).	File DSN

UNIX Client System Requirements

Linux Client System Requirements	
Hardware	Processor: Intel Processor - 2 cores or more Memory: 2 GB
Operating System	Any Linux OS which supports Linux Kernel 2.6.18 and above; for example, Fedora Release 6, RedHat Enterprise Linux Version 5, or SUSE Professional 10.2 Linux.
Free Hard Disk Space	50 MB
Software - ODBC Driver Manager	Any ODBC Driver Manager (tested with unixODBC)
Solaris Client System Requirements	
Hardware	Processor: UltraSPARC; Memory: 512 MB
Operating System	Sun OS 9 or above
Free Hard Disk Space	50 MB
Software - ODBC Driver Manager	Any ODBC Driver Manager (tested with unixODBC)
AIX Client System Requirements	
Hardware	Processor: IBM e-Server P-Series or RS/6000; Memory: 512 MB
Operating System	AIX 5.x Operating System: IBM AIX 5L Version 5.1, system maintenance level 2 (64-bit) or Version 5.2
Free Hard Disk Space	50 MB
Software - ODBC Driver Manager	Any ODBC Driver Manager (tested with unixODBC)
HP-UX Client System Requirements	
Hardware	Processor: PA-RISC, 512 MB
Operating System	HP-UX 11.0 (64-bit) or HP-UX V11.11i (64-bit)
Free Hard Disk Space	50MB
Software - ODBC Driver Manager	Any ODBC Driver Manager (tested with unixODBC)

All Unix/Linux systems must have libgcc runtime environments installed. If the 32bit client or 32bit JDBC server will be used on a 64bit system, then the 32bit libgcc runtime will need to be installed in addition to the 64bit libgcc runtime.

Important Note: Prior to installing the Linux/Unix client, CONNX must be installed on the Windows PC where the client installer is run from. Failure to do this will result in installation errors when installing the Linux/Unix client components.

User Application Requirements

In general, CONNX works with any ODBC-compliant application. Specifically, the programs listed here successfully utilize CONNX as a data access connectivity tool.

User Application Requirements

OLE DB - Windows only	OLE Automation (RPC) - Windows only	ODBC	JDBC	Application
	√			Any application that supports OLE automation
		√	√	Any JDBC-compliant application
√	□	√		Any ODBC- or OLE DB- compliant application
				Apache Web Server
√	√	√		Borland C++
√		√	√	Borland Delphi
		√	√	Borland JBuilder
	√	√		Cognos Impromptu
	√	√		Crystal Reports
	√	√		Dharma ODBC Integrator
□		√		GIS (Geographical Information Software)
√	√	√	√	Internet Information Server (IIS)
	√	√		JetForms
√	√	√		Microsoft Access
	√	√		Microsoft Excel (MSQuery)
		√		Microsoft SQL Server (linked server technology)
		√		Microsoft Transaction Server (MTS)
√	√	√		Microsoft Visual Basic
√	√	√		Microsoft Visual Basic for Applications (VBA)
√	√	√		Microsoft Visual C++, Microsoft Visual Studio
√	√	√		Microsoft Visual Studio .NET
		√	√	Netscape (iPlanet) Enterprise Server
	√	√		Oracle Developer/Designer 2000
		√		Oracle Discover
		√		Oracle Heterogeneous Services
√	√	√		PowerBuilder
√	√	√		Paradox for Windows
		√		Sagent

Step 1 - Installation Checklist and System Requirements

		√	√	Star Office
			√	Sun Forte
			√	Sun Netbeans
			√	Sun Netra Web Server
	√	√		Visual FoxPro for Windows

CONNX JDBC Client Requirements

Requirements	Minimum
JDK	6 or above
Free Hard Drive Space	500 MB

CONNX JDBC Server Requirements

Requirement	Minimum	Recommended
Processor	2 core processor	4 core processor
Memory	512 MB	1 GB
Free Hard Drive Space	500 MB	1 GB
Operating System	Windows Server 2012/Windows 10 or later AIX, UNIX, HPUX, Solaris, Linux/zLinux	Windows Server 2012/Windows 10 or later AIX, UNIX, HPUX, Solaris, Linux/zLinux
Network Connectivity	TCP/IP	TCP/IP
Software	64- and 32- bit libgcc runtime environment for 64bit Unix/Linux systems. 32- bit libgcc runtime environment for 32bit Unix/Linux systems.	

Default Port numbers

The following is a list of 32 and 64 bit components and the default ports they listen on:

Component Name	Default Port
32 bit Enterprise Server Service (Windows), CONNX Data Servers (IBM Mainframe, Unix, OpenVMS)	6500
64 bit Enterprise Server Service (Windows)	6502
32 bit JDBC Server (Windows, Unix)	7500
64 bit JDBC Server (Windows, Unix)	7502
License Server (Windows, Unix)	7501
CONNX Rest Server	9500

If CONNX is installed in an environment where a firewall is present, these ports need to be opened.

Step 2 - Install CONNX Administrator and CONNX Client

Windows Installation

General Information for Installation

This step assumes you have completed Step 1.

The CONNX Setup program installs [CONNX Administrator components](#) (the CONNX Data Dictionary), the CONNX client components (the CONNX ODBC driver and OLE server for Windows installations), the [OpenVMS server components](#) (for RMS, Rdb, and DBMS databases), [CONNX sample databases](#), the CONNX Enterprise Server Service, and CONNXStore on a client computer (see the CONNX User Reference Guide for more information on the CONNX Enterprise Server Service and CONNXStore [PostgreSQL]).

Installation of the OpenVMS server components places the data server files required for Rdb, DBMS, and RMS database connections on your computer. The files are then transferred by a network connection to their respective VAXServer or AlphaServer locations. The transfer of files to the VAXServer and AlphaServer is discussed in detail in [Step 4 - Install CONNX OpenVMS Server](#).

The administrator components enable the creation and modification of CONNX Data Dictionary (CDD) files, which store and maintain database table definitions. The administrator components are a Microsoft Windows-based program and must be installed on at least one network machine running a Windows operating system.

A CONNX license file is required for installation of the administrator components. The license file gives a site administrator control over which users can install the administrator component.

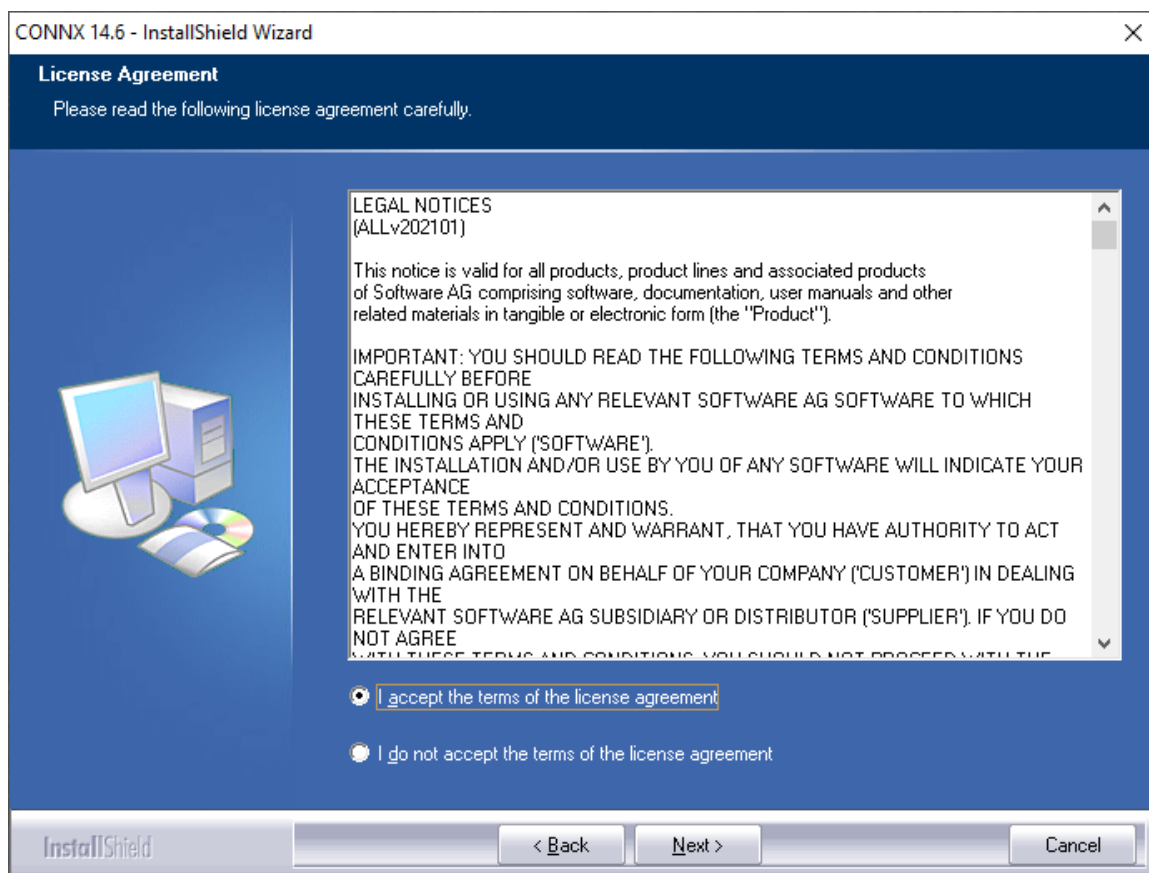
Note: Although the administrator components can be installed on several computers, the CDD files should be placed in a single shared network server location. This enables a complete listing of CDD files created by the administrators to be made available to more than one user.

CONNX Installation Procedure

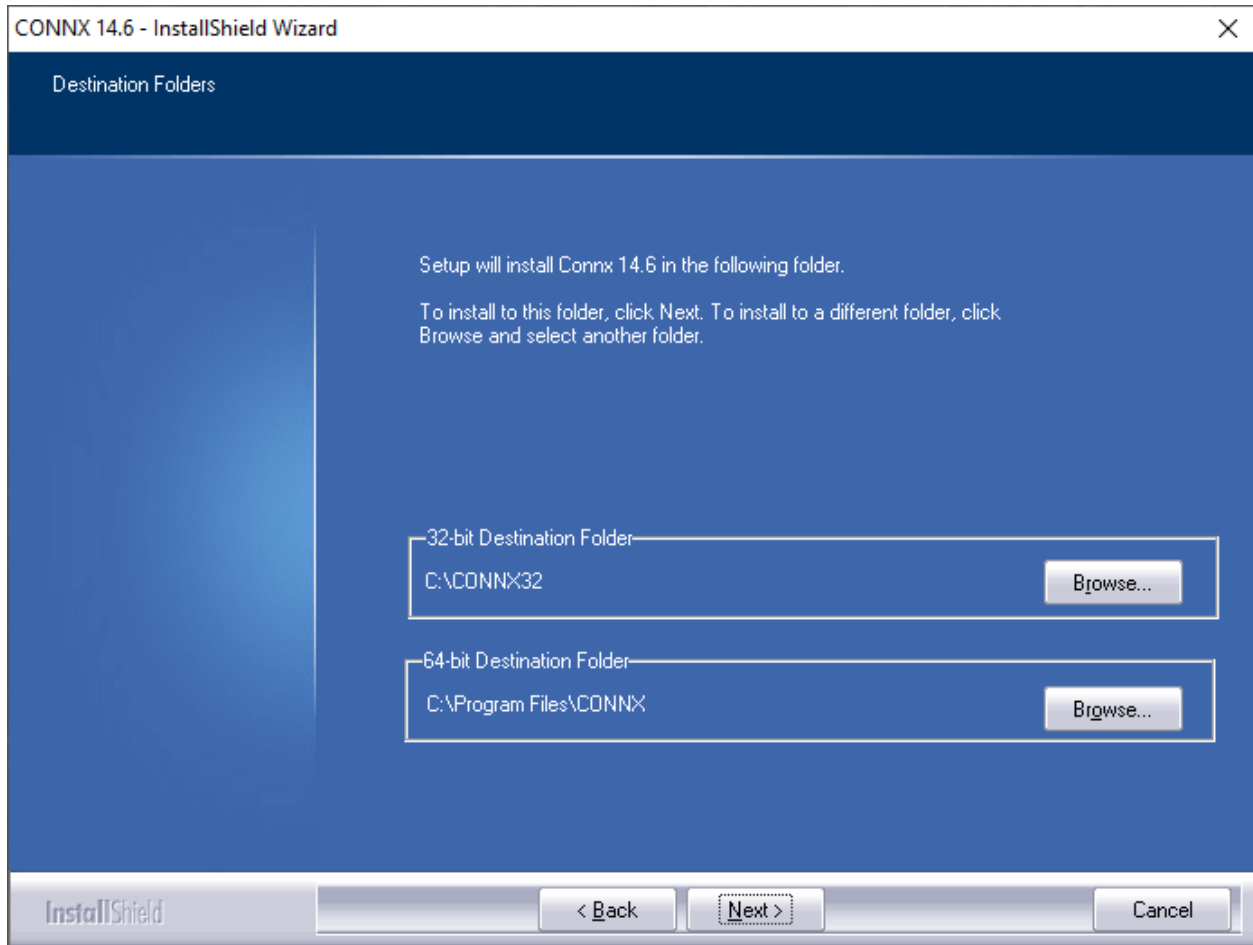
The following instructions describe installation procedures for CONNX Administrator and CONNX client components.

To install CONNX Administrator and CONNX client components

1. **If you have not done so already**, open the email you received containing the CONNX installation information and copy the license files from the email to a shared location that all CONNX administrative computers can access. The installation process will ask where the license files are located in a later step; make a note of where you have saved the files.
2. Download and run the installation kit from Empower or use the download links if any were provided during your maintenance renewal or evaluation period.

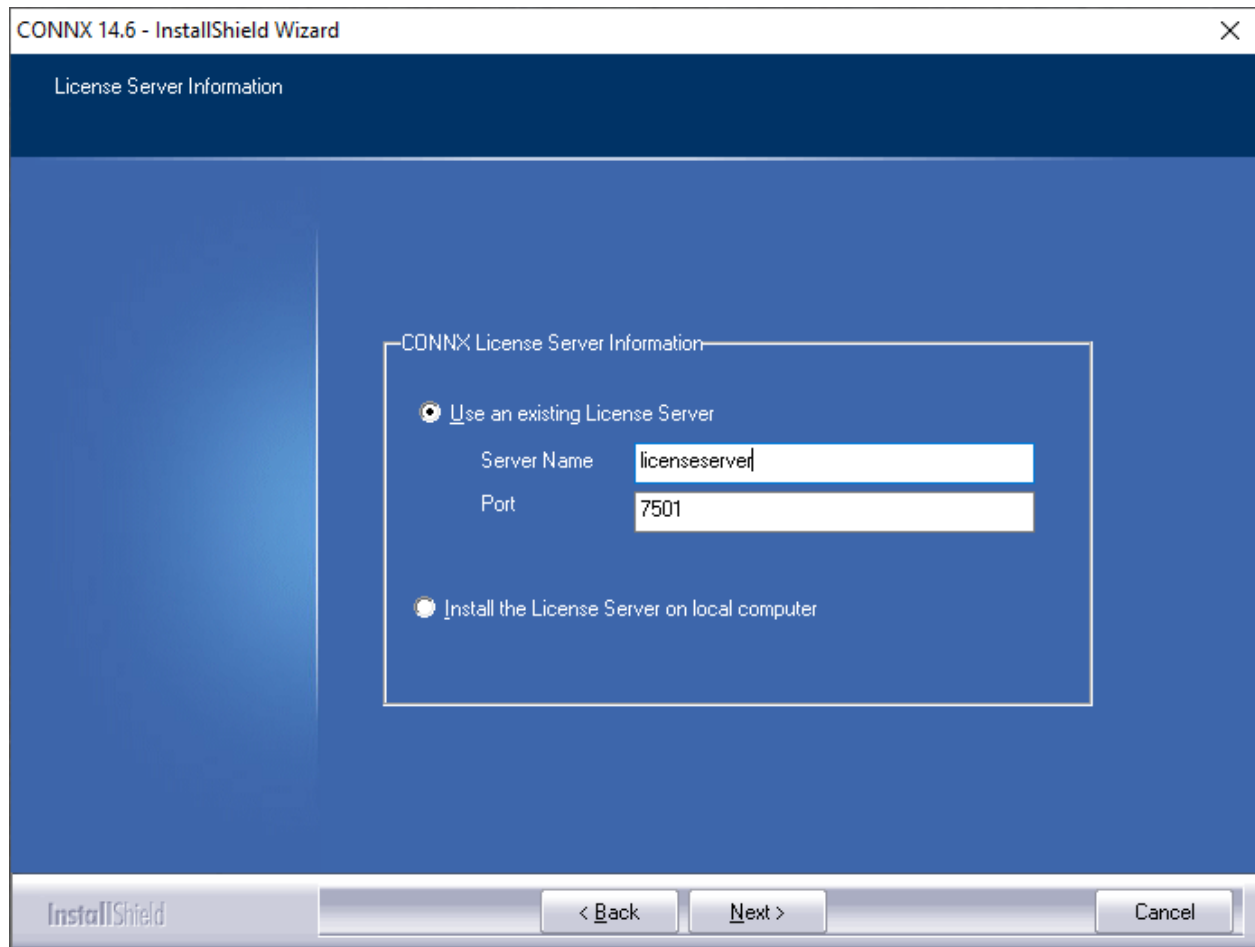


3. Click **"I accept the terms of the license agreement"** radio button.. The **Destination Folders** window appears. On 64-bit systems, you have the option of selecting the location of the 32-bit, and 64-bit components separately. On 32-bit systems, you will only have the option for selecting the location of 32-bit components.

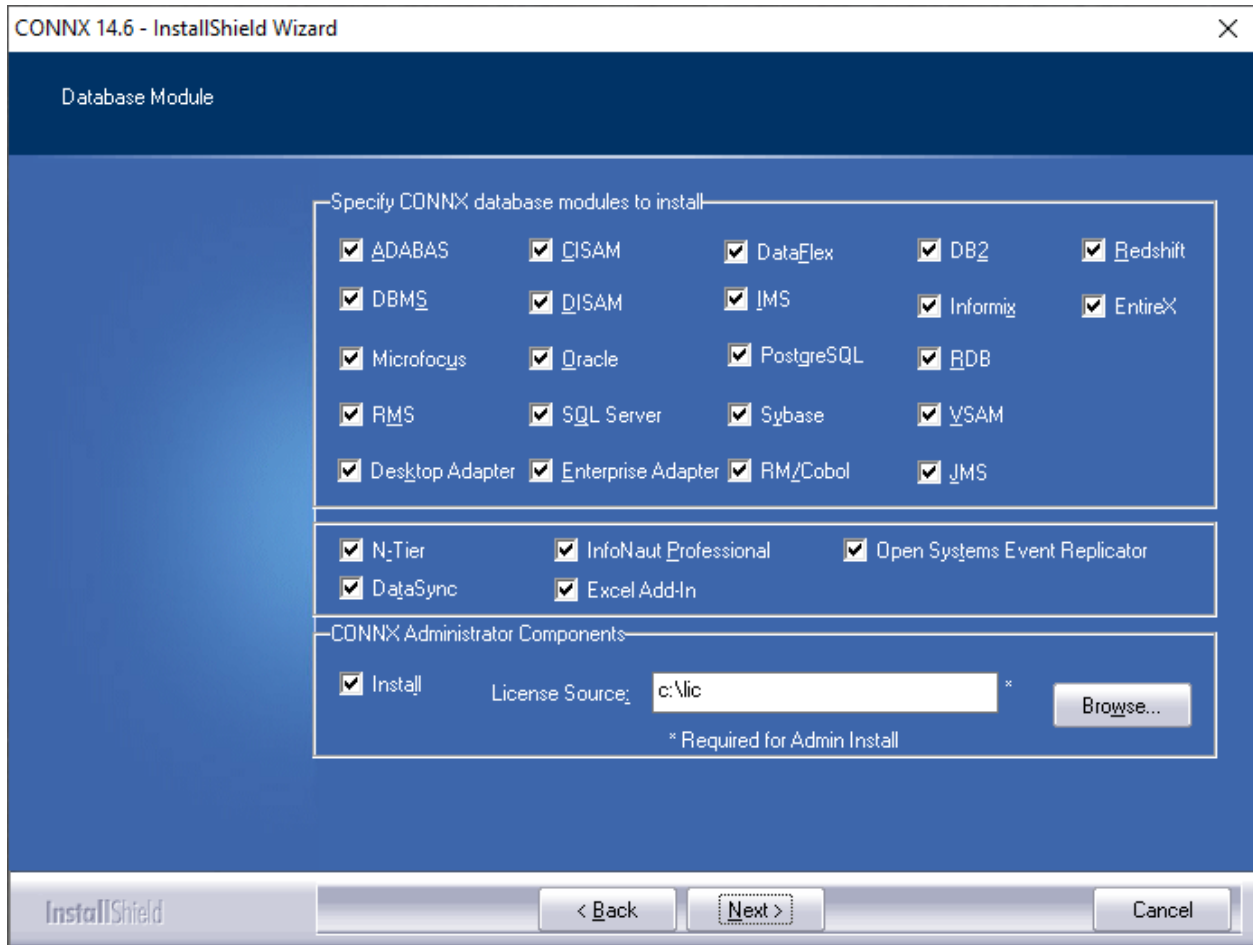


4. Click **Next**. The **License Server Information** window appears. If you already have a CONNX license server installed on your network, specify the correct server name and port number. The default port number for the CONNX license server is 7501. If you want to install the license server locally on this computer, select the appropriate radio button

Warning: The **CONNX License Server** needs to be installed on the network **one time only**, regardless of whether the user is on the LAN or WAN or connected via VPN. If the CONNX License Server has already been installed, simply specify the name of the license server and the port number (default is 7501).



5. Click **Next**. The **Database Modules** window appears.



6. In **Specify CONNX database modules to install**, click the database modules you have purchased or are evaluating. **You must have a license file for each database module you select.**

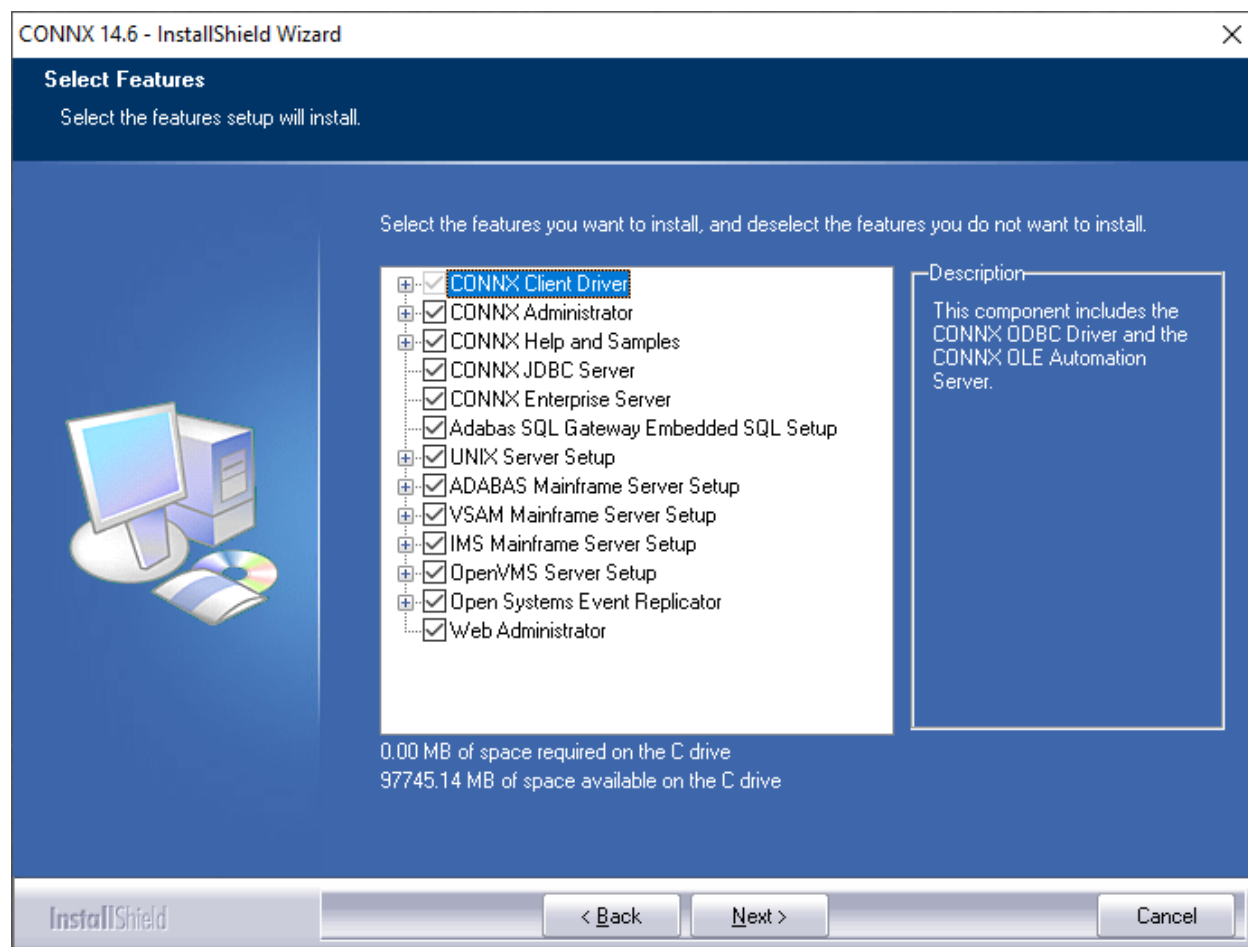
Select **DataSync**, **InfoNaut Professional** or **N-Tier** only if you purchased or are evaluating these products. For more information on any of these products, contact your CONNX sales representative.

Note: InfoNaut Lite is included with all CONNX licenses. Select **InfoNaut Professional** only if you purchased or are evaluating this product.

7. In **CONNX Administrative Components**, if you are installing CONNX on an administrative computer, click **Install** and enter the file path to that folder containing the license files in **License Source**. This will install the CONNX Data Dictionary tool and the License Administrator program.

If you are not installing CONNX on an administrative computer, clear **Install**.

8. Click **Next**. The Select Components window appears.



9. Select the check boxes beside the CONNX components you want to install. A description of each component appears. Click the **Change** button to alter the sub-components.

Note: You can save drive space by clearing the unused sub-component check boxes. **CONNX Components and Sub-Components** below contains a list of the CONNX components, sub-components, and component descriptions.

CONNX Components and Sub-Components

Component	Description	Subcomponents
CONNX Client Driver	Required on all CONNX client machines.	InfoNaut Query Tool CONNX ODBC Driver CONNX OLE Automation Object CONNX JDBC Components
CONNX UNIX Client Driver	Required on all UNIX based CONNX client machines.	CONNX ODBC Driver CONNX JDBC Components
OpenVMS Server Components	Required for Rdb, RMS, and DBMS databases. Must be installed on at least one administrator machine, and then transferred to an OpenVMS	CONNX OpenVMS Setup Alpha components VAX components

	database server.	Itanium components (RMS only)
IMS Mainframe Server Components	Required for access of IMS data on the mainframe.	Setup Utility
VSAM Mainframe Server Components	Required for access of VSAM data on the mainframe.	Setup Utility
CONNX JDBC Server	Must be installed on at least one machine if you use JDBC.	CONNX JDBC Binaries
UNIX Server Setup	Required for access of Adabas, C-ISAM, DISAM, Micro Focus and RM/Cobol data on Linux/UNIX.	Setup Utility for Adabas, C-ISAM, DISAM, Micro Focus and RM/Cobol
CONNX Help and Samples	For users who want to test different types of sample databases and front-end applications. Help files can be installed on both admin and client machines.	CONNX Help Files Microsoft® Access® Sample CONNX OLE RPC Test Drive for VMS Data Dictionary Viewer InfoNaut Lite
CONNXStore Database Server	Native CONNXStore database server (PostgreSQL for windows). Required for DataSync. Note: If DataSync was checked on the Database Module screen, CONNXStore will be installed and there will be no check box option for it on the component screen.	
CONNX Enterprise Server Service	Allows data sources hosted on Windows to be accessed from other machines and operating systems	
Adabas Mainframe Components	Required for access of Adabas data on the mainframe.	Setup Utility
InstantdbSync (Open Systems Event Replicator for Adabas customers) components	Allows selection of individual InstantdbSync components.	Installer for UNIX components

11. If you are installing the Adabas Mainframe Components, scroll down to view the ADABAS Mainframe Server Setup check box. Select the check box if you are installing or updating the server components on the mainframe. This step only has to be done once or when the server changes. If you will be installing multiple clients later, it is not necessary to check the check box during installation. For information on all Adabas installation configurations, see [Recommended Configurations for the Adabas SQL Gateway \(CONNX for Adabas\)](#).

Note: Install the CONNX client components on each computer requiring data access. See [Silent CONNX Client Installation](#). You do not have to install the administrator component and JDBC server components on machines requiring data access only.

Warning: For administrator machines, ensure that both the CONNX client driver and the CONNX administrator components are installed. The OpenVMS server components required for Rdb, DBMS, and

RMS databases must be installed on at least one administrator machine and then transferred to the OpenVMS database server machine. The server components need not be installed on client machines. See [CONNX OpenVMS Installation Procedure](#) for information on transferring the server components to the server. For client machines requiring data access only, select only the CONNX Client Driver component in the Select Components window.

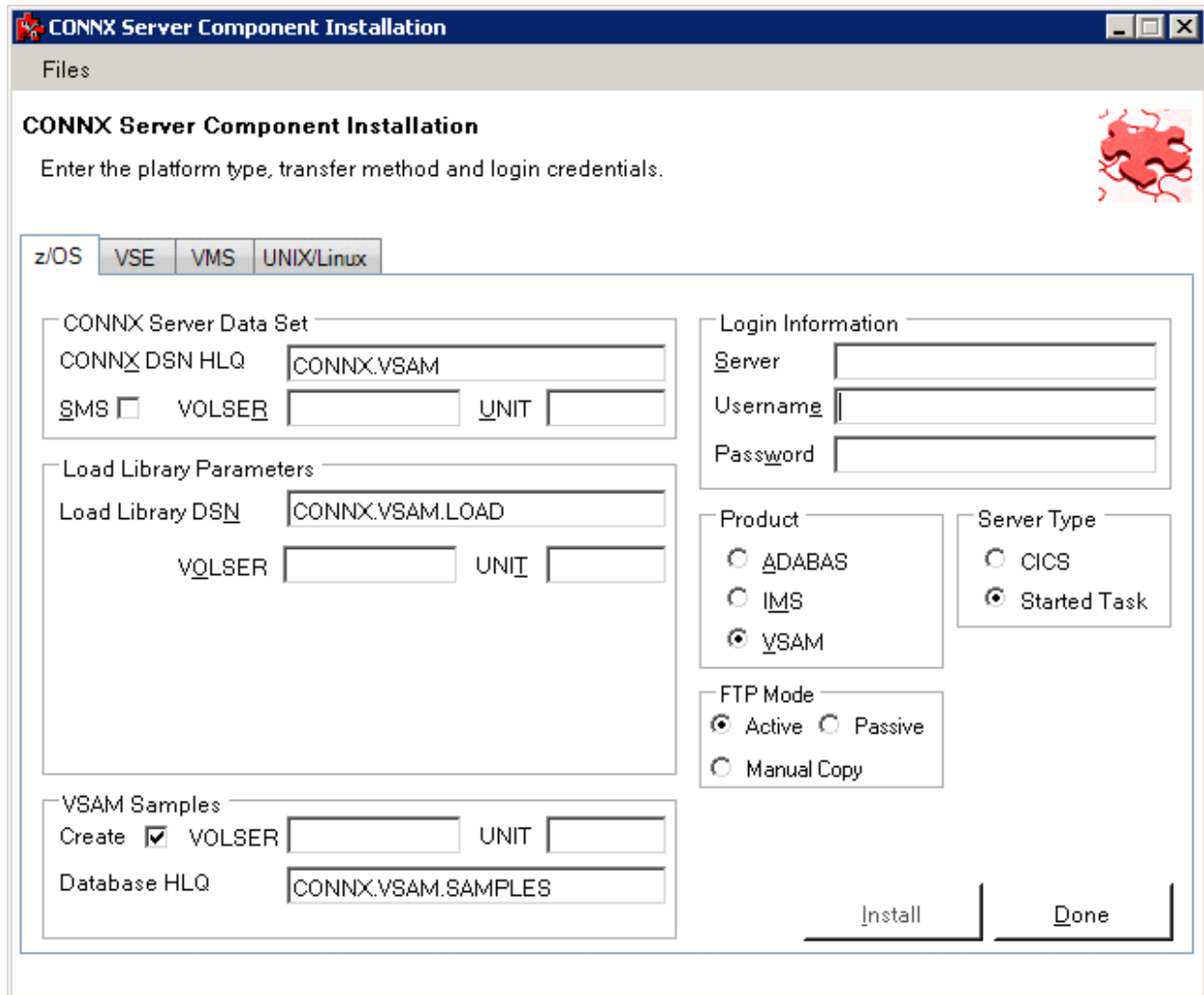
Warning: Do not select UNIX Server Setup if you are using C-ISAM, DISAM, or Micro Focus on a Windows platform.

Warning: Do not select Adabas Mainframe Components if you are using Adabas on a Windows platform.

12. If you have selected the CONNX JDBC Server Service, the Enterprise Server Service, the CONNX Store Database Server Service, InstantdbSync or Install CONNX License server on local computer, the **CONNX Service Account Selection** dialog box appears. To have the services run as a specific domain user, Select **This account** and type a domain or user name in the **Domain\UserName** text box in the format shown. Type a password in the **Password** text box. To have the services run under the local system account, select **Local System account**.

The screenshot shows a dialog box titled "CONNX 14.6 - InstallShield Wizard" with a close button (X) in the top right corner. The main title bar is "CONNX service account selection". Inside the dialog, there are two radio buttons: "This account" (which is selected) and "Local System account". Below the "This account" radio button, there are three text input fields: "Domain\UserName:" with the text "domain\user", "Password:" with masked characters (dots), and "Confirm Password:" with masked characters (dots). A "Next >" button is located at the bottom right of the dialog.

13. Click **Next** and the installer will copy the files. When the copy process has finished, an information message box appears stating that the CONNX Server Setup programs are set to install the server components that you have selected. Click **OK**. The **CONNX Server Component Installation** window appears. See [Step4 - Install CONNX OpenVMS Server](#), [Step 5 - Install IBM Mainframe-Compatible Servers](#) (includes Adabas installation for these servers), [Step 6 - Install Adabas SQL Gateway](#) for Adabas UNIX installation), or [Step 7 - Install C-ISAM, D-ISAM and Micro Focus Servers component on a UNIX machine](#) for more information on installing these server components.



CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS **VSE** VMS UNIX/Linux

CONNX Server Data Set
 CONNX DSN HLQ
 SMS ☐ VOLSER UNIT

Load Library Parameters
 Load Library DSN
 VOLSER UNIT

VSAM Samples
 Create ☒ VOLSER UNIT
 Database HLQ

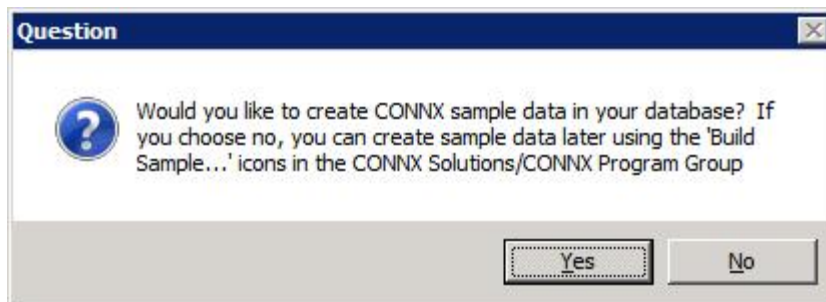
Login Information
 Server
 Username
 Password

Product
☐ ADABAS
☐ IMS
☒ VSAM

Server Type
☐ CICS
☒ Started Task

FTP Mode
☒ Active ☐ Passive
☐ Manual Copy

14. If you selected Oracle, DB2, C-ISAM, Informix, Sybase, ODBC, or OLE DB data sources in the Database Modules window, a dialog box appears asking whether you want to install sample database table.

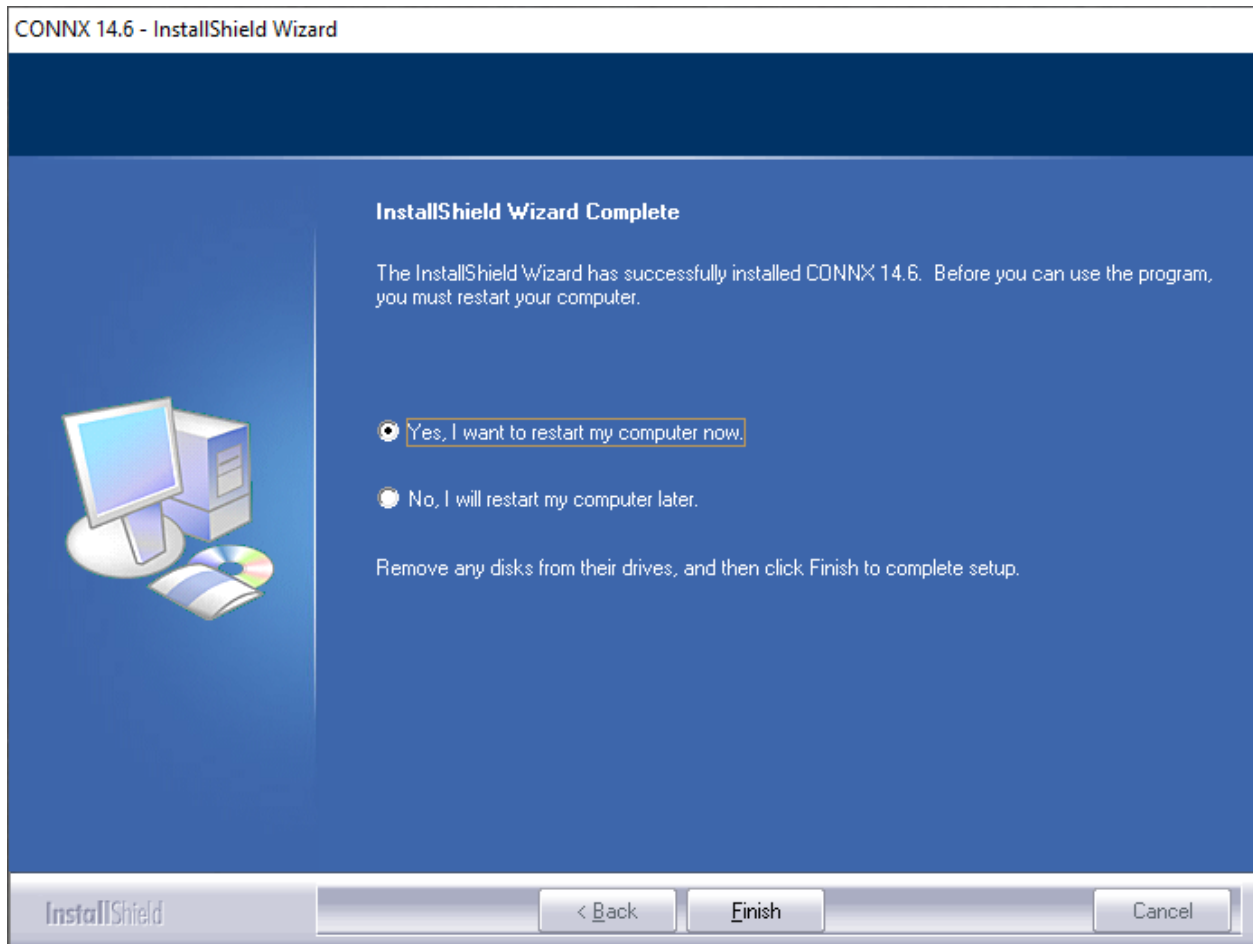


Question

Would you like to create CONNX sample data in your database? If you choose no, you can create sample data later using the 'Build Sample...' icons in the CONNX Solutions/CONNX Program Group

Click **Yes** to continue with sample table installation through **CONNX Sample Database Logon** dialog boxes. Click **No** to continue with CONNX installation. You can install sample tables for each of the listed databases at any time using the **Build Sample Database** icons available in your program group.

15. This completes the preliminary CONNX installation procedures. **If you are prompted to reboot after the installation finishes, please do so.**



UNIX Installation

General Information for Installation - UNIX System

This step copies the CONNX client components from a client PC to an UNIX target (AIX, Sun, Solaris, HP-UX).

Important: All UNIX instructions must be entered as lowercase characters.

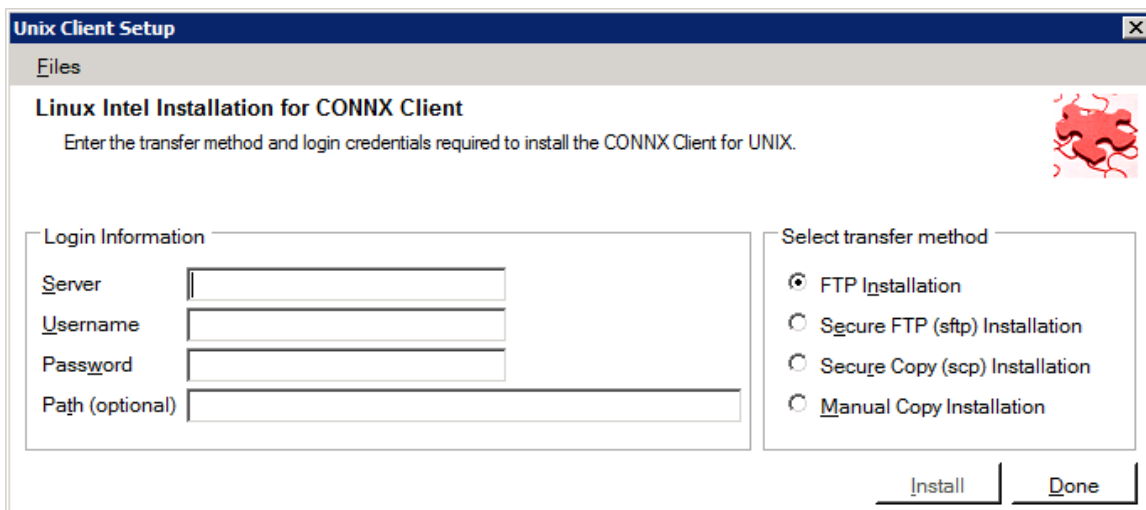
There are separate installation packages for each of the UNIX target systems.

Important: The CONNX Administrator component must be installed on the client PC.

CONNX Installation Procedure - UNIX

The installation instructions for your UNIX system can be found on the Empower web site.

1. Download the appropriate UNIX installation package from Empower to your computer.
2. Double-click on the installation package icon to launch the Install Shield Wizard for CONNX Client.

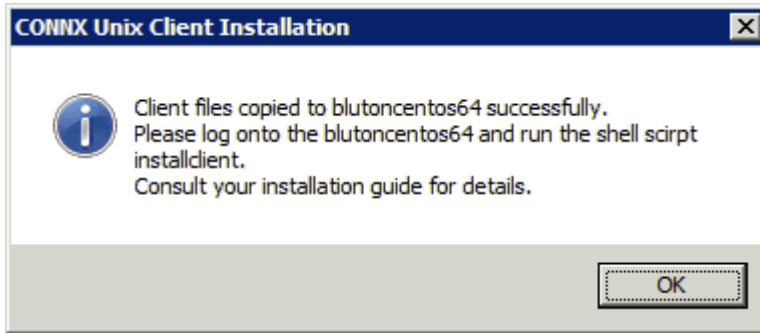


3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
 4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the UNIXCLIENT\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 11 below.
 5. Select the desired transfer method (FTP, SFTP or SCP)
 6. In the **Server** text box, enter the **TCP/IP host name or address** for your system platform.
 7. In the **User ID** text box, enter a privileged user account name.
 8. In the **Password** text box, enter a user account password. (The password appears as *****).
- Note:** **Server**, **User ID** and **Password** are required fields.
9. By default, the CONNX client data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **UNIX Directory** text box. If the specified directory does not exist, it will be created.
 10. Click the **Install** button. This action starts an FTP, SFTP or SCP session and copies the selected components to the target server. The progress of the copy will be displayed in the status pane on the installer..
 11. An ftp window appears saying the files were successfully copied to the system.

12. If the CONNX client is not installed, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.

Note: If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.

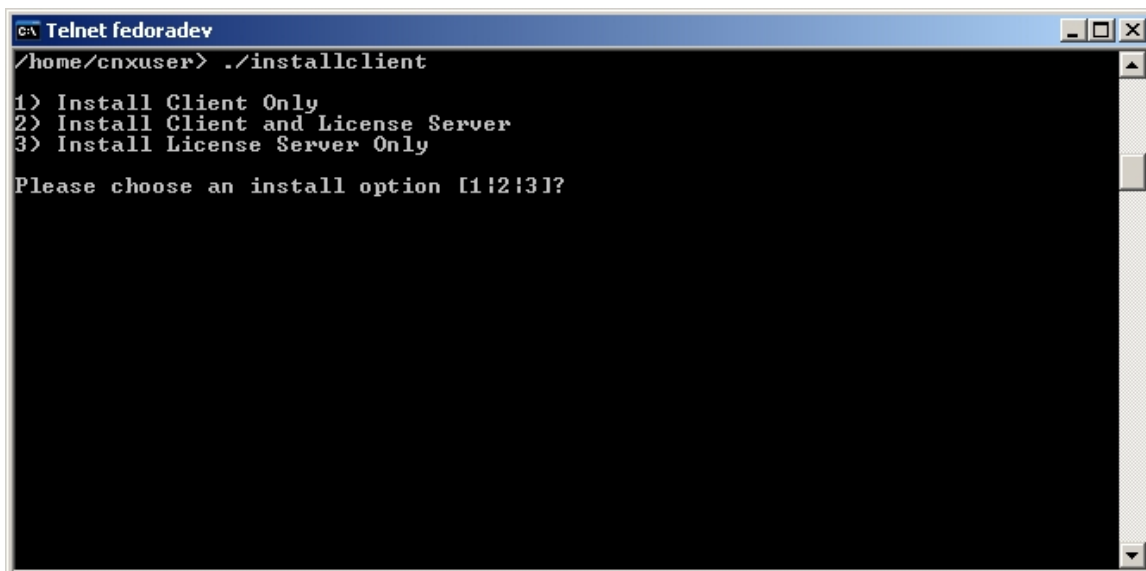
13. Once the files are copied, a dialog box appears asking you to log on to your system and run the installclient shell script.



14. To run the installclient shell script, start a terminal emulator on the UNIX server and type the appropriate user name and password.

15. Type `./installclient` and then press <Enter>.

You will see a screen that gives three options. With SQL Gateway a License server is required in order to use the product. In many circumstances when a Unix Client is desired, it might be desirable to have the license server on the same machine. If this is the case, you would select option 2. If you wish to use an existing License Server, that might be running on a different machine, then select option 1. If you wish to install the License Server only, then select option 3. and then press <Enter>. Press <Enter> to install to current directory or enter a new install path at the prompt. The contents of the compressed files are unzipped to a CONNX directory in the install directory.



```

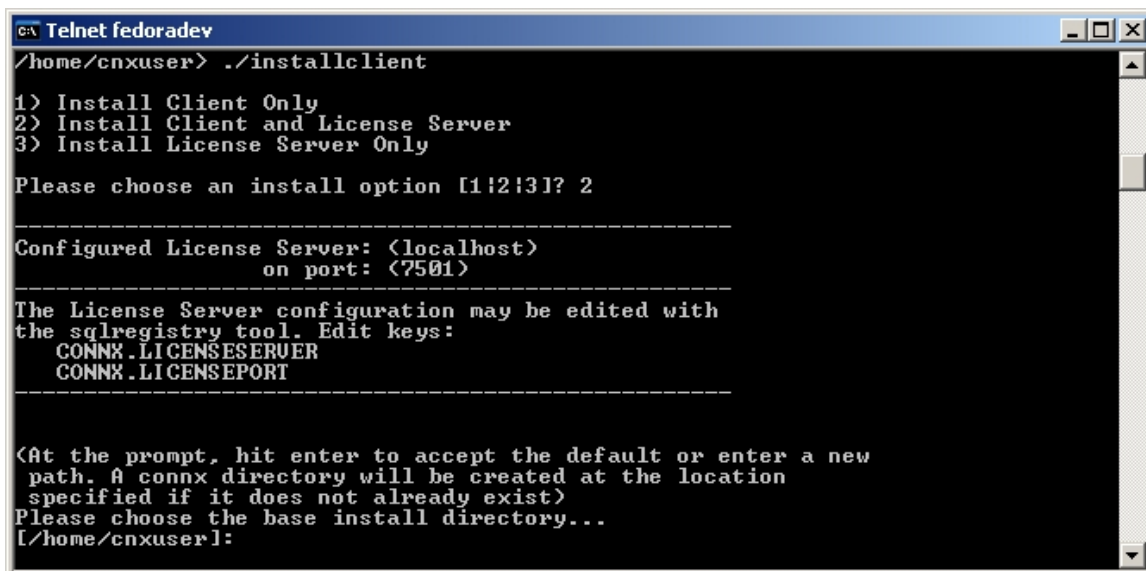
CA Telnet fedoradev
/home/cnxuser> ./installclient

1> Install Client Only
2> Install Client and License Server
3> Install License Server Only

Please choose an install option [1!2!3]?

```

If option 2 is selected, you will notice that it will try to install the License Server on your UNIX Machine as localhost with an address of 7501. After this installation script has been run, the License Server will be automatically invoked. In the future, you can start/stop or check the status of the License Server by invoking the script, `licenseserver`. This script is in the CONNX folder. The parameters to this script are `[start|stop|status]`. So to start the license server, you would type, `./licenseserver start`. You might consider adding this to your startup script in order to have the license server running when the UNIX machine is running. The screen below shows the case where option 2 is selected.



```

CA Telnet fedoradev
/home/cnxuser> ./installclient

1> Install Client Only
2> Install Client and License Server
3> Install License Server Only

Please choose an install option [1!2!3]? 2

-----
Configured License Server: <localhost>
                        on port: <7501>
-----

The License Server configuration may be edited with
the sqlregistry tool. Edit keys:
  CONNX.LICENSESERVER
  CONNX.LICENSEPORT
-----

<At the prompt, hit enter to accept the default or enter a new
path. A connx directory will be created at the location
specified if it does not already exist>
Please choose the base install directory...
[/home/cnxuser]:

```

15. The final output should read:

CONNX UNIX Client Installation Complete.

16. To run the CONNX client, add the correct location of the registry files (64 and 32 bit) and the environment variables (`CONNXREGISTRY` and `CONNXREGISTRY64`) pointing to the correct location of the registry files to the **env** list. For example, add the entries:

```
export CONNXREGISTRY=<LocationOfRegistry>/connxreg.db  
export CONNXREGISTRY64=<LocationOfRegistry>/connxreg64.db
```

to the startup profile of any user that runs the CONNX client application. The location of the registry is specified during the client installation or by default is the same folder in which the CONNX client is installed.

17. Close the telnet server session.
18. Click the **Done** button.
Important: All UNIX instructions must be entered as lowercase characters.
19. If you installing a Unix system with a ODBC driver manager, follow the instructions in Completing CONNX Installation for UNIX ODBC Drivers.
20. If you are installing a Unix system with a JDBC server, follow the instructions in Completing CONNX Installation for JDBC Servers.
21. The first part of the UNIX CONNX installation process is complete. [Refer to the appropriate Post-Installation Checklist](#) to complete any additional requirements for each database you install.

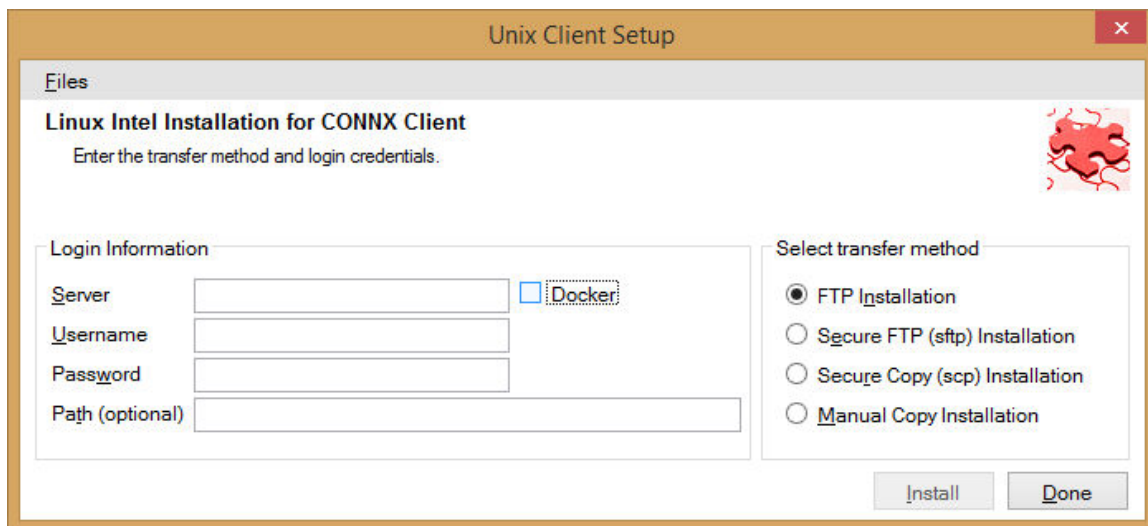
CONNX Installation Procedure - Docker

CONNX provides Docker scripts that facilitate the creation of Docker images for the CONNX components on the Linux operating system.

Docker scripts are provided for the CONNX Data Server, the CONNX License server, and the CONNX JDBC server. The follow section will over the JDBC server and the License server.

The installation instructions for your Linux system can be found on the Empower web site.

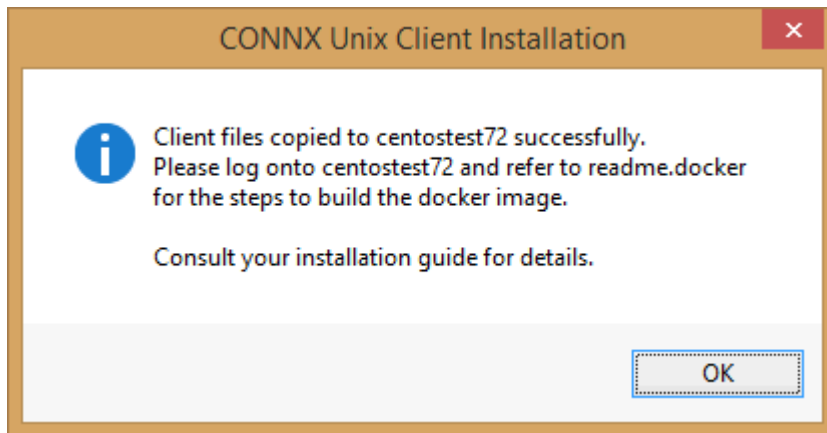
1. Download the appropriate Linux installation package from Empower to your computer.
2. Double-click on the installation package icon to launch the Install Shield Wizard for CONNX Client.



3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the UNIXCLIENT\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 11 below.
5. Select the **Docker** checkbox.
6. Select the desired transfer method (FTP, SFTP or SCP)
7. In the **Server** text box, enter the **TCP/IP host name or address** for your system platform.
8. In the **User ID** text box, enter a privileged user account name.
9. In the **Password** text box, enter a user account password. (The password appears as *****).

Note: **Server**, **User ID** and **Password** are required fields.

10. By default, the CONNX client data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **UNIX Directory** text box. If the specified directory does not exist, it will be created.
 11. Click the **Install** button. This action starts an FTP, SFTP or SCP session and copies the selected components to the target server. The progress of the copy will be displayed in the status pane on the installer..
 12. An ftp window appears saying the files were successfully copied to the system.
 13. If the CONNX client is not installed, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.
- Note:** If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.
14. Once the files are copied, a dialog box appears asking you to log on to your system and run the installclient shell script.



15. To build the Docker image, start a terminal emulator on the Linux server and type the appropriate user name and password.
16. Navigate to the directory you specified in the windows file transfer GUI. This directory will contain several files, including two template Docker files.

Dockerfile.jdbcserver is the template for building a Docker image for the JDBC Server. This script should be modified to include your custom configuration settings and DSN entries. The CDDs used by the docker image are externalized, and should be placed in the host docker system. Please refer to Completing CONNX Installation for JDBC Servers for the specific commands to add to the docker script for creating DSNs. Examples are also provided directly in the docker script.
17. After the Dockerfile.jdbcserver has been updated to include DSN and other configuration settings, build a docker image for the CONNX license server, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker build -t connxXXXjdbcserver . -f./Dockerfile.jdbcserver
```

Example:

```
docker build -t connx138jdbcserver . -f./Dockerfile.jdbcserver
```

18. The final output should read:

Successfully tagged connxXXXjdbcserver:latest.

19. The next step is to build a docker image for the CONNX license server. No modification of the supplied docker script should be required.

Run the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker build -t connxXXXlicenseserver . -f./Dockerfile.licenseserver
```

Example:

```
docker build -t connx138licenseserver . -f./Dockerfile.licenseserver
```

20. The final output should read:

Successfully tagged connxXXXlicenseserver:latest.

21. To run the CONNX JDBC docker image, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxclient -v <Config file location>:/config:z -p  
<external port>:7502 connxXXXjdbcserver
```

Example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxclient -v /home/cnxuser/config:/config:z -p  
20002:7502 connx138jdbcserver
```

The <Config file location> should contain the CONNXREG64.db file which holds the CONNX configuration information for the JDBC server.

This directory should also contain your CONNX CDD files.

Ensure that the directory on the host system has an owner and group of sagadmin with an id of 1724.

The <external port> should be the port number for connecting to the JDBC server.

22. To run the CONNX license server docker image, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxlicense -v <license file location>:/license:z -p  
<external port>:7501 connxXXXlicenseserver
```

Example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxclient -v /home/cnxuser/license:/license:z -p
```

20001:7501 connx138licenseserver

The <license file location> should contain your CONNX license database - it will be created and maintained by the license server, and it is external from the docker image.

Ensure that the directory on the host system has an owner and group of sagadmin with an id of 1724.

The <external port> should be the port number for connecting to the license server.

23. The first part of the Docker CONNX installation process is complete. [Refer to the appropriate Post-Installation Checklist](#) to complete any additional requirements for each database you install.

24. Verifying the Build:

Please refer to the following docker command reference to manage and verify the image and container.

Show the available docker images:
docker images

Start the docker image - for example:
docker run -d -e "ACCEPT_EULA=Y" --name cnxclient -v /home/cnxuser/config:/config:z -p 20002:7502 connx138jdbcserver

Show the log:
docker logs cnxclient

Show the containers:
docker ps -a

Stop the container:
docker stop cnxclient

Delete the container:
docker rm cnxclient

Remove the image:
docker image rm connx138jdbcserver

Completing CONNX Installation for UNIX ODBC Drivers

This continues the UNIX installation for systems with ODBC drivers. The installation started in CONNX Installation Procedure - UNIX.

Note: When using the ODBC interface, you must install an ODBC driver manager. There are several driver managers available and CONNX Solutions, Inc. does not endorse one over the other. The examples below are based on the Open Source ODBC Driver Manager unixODBC. When installing this or any other driver manager, please refer to the installation instructions for the specific product.

Note: All UNIX instructions must be entered as lowercase characters.

1. Create the connxdriver.template file.

Once CONNX has been installed on the UNIX system, it can be accessed through an ODBC driver manager. The ODBC driver manager defines how to register the CONNX driver and/or the name of your data source (DSN).

The following example gives the information (with suggested values) needed to register CONNX to the unixODBC driver manager. (Visit <http://www.unixODBC.org/> for download information.) Enter this information into a text file using your favorite text editor:

[CNXODBC] The CONNX driver name, including the brackets. We recommend you use CNXODBC as the driver name.

Description = CONNX ODBC Driver The description is treated as a comment; it is not mandatory.

Driver = /installdirectory/connx/libxx/libconnx32.so

Setup = /installdirectory/connx/libxx/libconnx32.so

installdirectory is the location of the CONNX UNIX client.

libxx is the 32bit or 64bit lib directory. For 64bit, the value will be lib64. For 32bit, the value will be lib32.

The extension of the libconnx32 file changes depending on the platform. The example shows **.so** for Linux and Sun; other extensions are **.a** for AIX, and **.sl** for HP-UX.

Threading = 2 Gives the number of threads for multitasking. Minimum multitasking value is 2.

Save this information in the connxdriver.template file, then use the command: **odbcinst -i -d -f connxdriver.template** to register the driver with unixODBC.

2. Create the connxdsn.template file:

The following example gives the information needed to register a DSN with unixODBC. Enter this information into a text file using your favorite text editor:

[CONNXDSN] The data source name, including the brackets.

Description = CONNX Samples Data Source The description is treated as a comment; it is not mandatory.

Driver = [CNXODBC] Since you have registered the driver in the previous step, use the format Driver=[drivername]

DATADictionary = /installdirectory/connx/samples.cdd This is the path and filename for the data dictionary. This file cannot be created or edited on a UNIX environment. Create or edit this file on Windows, then ftp it to your Unix system.

APPLICATION = <application or database> Your application or database type, such as Adabas or CISAM.

If you are running multiple applications or database types, leave this blank.

Trace = No Turn trace on (yes) for debugging.

TraceFile = sql.log Ignored if Trace = No; otherwise the trace output goes here.

UserName = CONNX user ID

Password = CONNX password

The UserName and Password can be left blank. If they are blank, when you connect to CONNX, you will be prompted for that information.

Save this information in the connxdsn.template file, and then use the command: **odbcinst -i -s -l -f connxdsn.template** to register the driver with UnixODBC.

3. Use a query program to connect to the data source using the newly created DSN to retrieve data.
4. The iSQL utility that enables you to test connections is installed with unixODBC. After you create a DSN, in preparation for connecting to your data source, run **isql** from a command prompt. The format for running isql is:

isql {cmd options} [dsn name] {username} {password}

For example:

isql -v connxsample connx connx

5. If the connection is successful, a "Connected!" message appears at the bottom of the screen. If the connection is not successful, the "[ISQL]ERROR" message appears. For more information on what to do if the connection fails, use the **-v** option (verbose mode) to analyze the problem.

Completing CONNX Installation for UNIX JDBC Servers

This contains the UNIX installation systems with for JDBC servers. The installation started in CONNX Installation Procedure - UNIX.

1. If necessary, edit the UNIX JDBC server startup procedure.
2. The default JDBC port is 7500. If necessary, use the CONNX SQL Registry to change CONNX.JDBC.PORT from 7500 to the correct port number:
 - A. At the command prompt, enter **./sqlregistry**
 - B. Select option **2** to create a registry key and value
 - C. Enter **CONNX.JDBC.PORT**
 - D. Select option **0** for an integer value
 - E. Enter the correct port number
 - F. Enter **5** to exit the the **./sqlregistry** program
3. Create the UNIX JDBC DSNs
 - A. At the command prompt, enter **./createJDBCDSN**
 - B. At the command prompt, use the format **<DSN> <CDD> [comment]** where
 - **DSN** = top level name of the CONNX JDBC data source name
 - **CDD** = the fully qualified path to the CONNX CDD. The CDD must be located on a UNIX system.
4. Start the UNIX JDBC Server. If the installation is successful, a **JDBC server started successfully** message appears.

CONNX Sample Database Tables

Creating CONNX Sample Database Tables

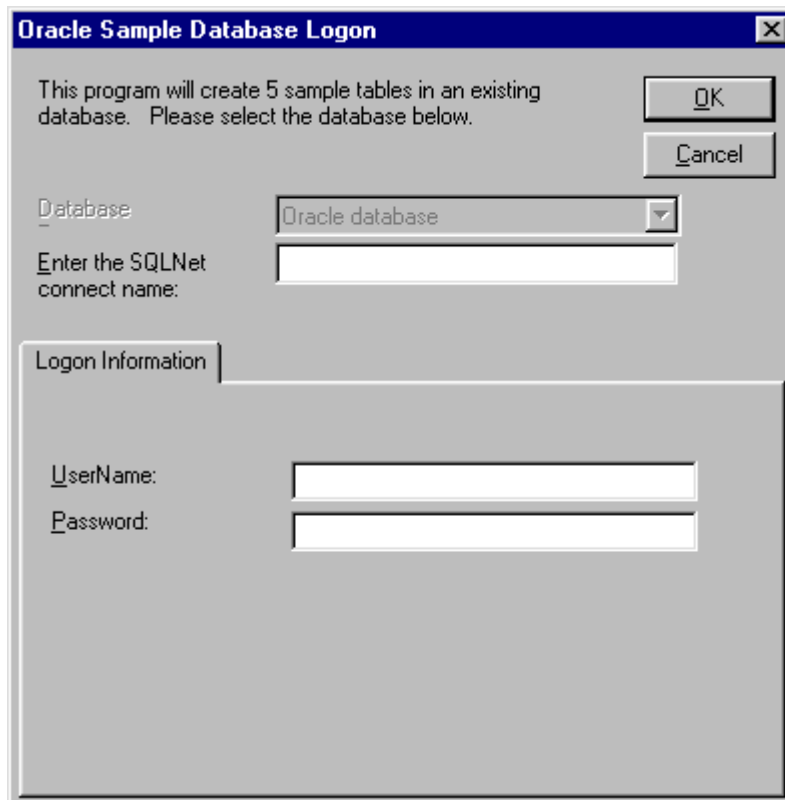
If you selected Oracle, DB2, or Enterprise or Desktop OLEDB/ODBC Adapter data sources in the Database Modules window, CONNX Sample Database Logon dialog boxes appear for each selected data source. Refer to the next sections for information on how to install the sample tables.

Sample database tables can be created either during installation or after installation. During installation, a dialog box appears asking whether you want to install sample database tables. Click Yes to continue with sample table installation through CONNX sample database logon dialog boxes. Click No to continue with CONNX installation. You can install sample tables at any time using the Build Sample Database icons available in your program group.

To create Oracle sample database tables

The following procedure runs a script designed to insert CONNX sample data into an Oracle database installed on an Oracle server.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build Oracle Sample Data**. The Oracle Sample Database Logon dialog box appears.



The dialog box is titled "Oracle Sample Database Logon" and contains the following elements:

- A message: "This program will create 5 sample tables in an existing database. Please select the database below."
- Buttons: "OK" and "Cancel" in the top right corner.
- A "Database" label followed by a dropdown menu showing "Oracle database".
- A label "Enter the SQLNet connect name:" followed by an empty text input field.
- A tabbed section titled "Logon Information" containing:
 - A "UserName:" label followed by an empty text input field.
 - A "Password:" label followed by an empty text input field.

Important: You must have write privileges in order to use the Oracle sample tables.

2. Confirm that the Oracle Client is installed on the client machine, and then type your SQLNet Alias and your Oracle user name and password.
3. After installation is complete, click the **OK** button to return to the final installation procedures. You can refer to the samples in the post-installation stage by opening the CONNX Data Dictionary Manager window.

To build CONNX for DB2 sample tables on the target host via TCP/IP

These tables can be built during installation of the CONNX DB2 client, or later via a command prompt (MS-DOS). The following procedures encompass both methods. See **Step 8 - DB2 Host Configuration** for configuration information before attempting to install the sample database tables.

Note: Refer to the CONNX User Reference Guide online or on the distribution media for more detailed information about logging on and dialog box text boxes and options.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build DB2 Sample Data**. You may also start an MS-DOS command prompt, and then go to the CONNX installation directory. Select the **Utils** subdirectory, for example, **C:\connx32\Utils**. From the MS-DOS command prompt start the CONNX sample tables application by typing **start sampgen db2**
2. The first **DB2 Sample Database Logon** dialog box appears after the prompt, before installation completes, or if you selected **Build DB2 Sample Data** in the program group.

In this example, the RDBNAME is an example; substitute the RDBNAME or location name of the target DB2 host.

3. Click the **Settings** tab.

The owner list shows a sample schema name = CONNX. All DB2 databases require three-part names for SQL objects, such as tables and views. A fully qualified table name is as follows:

Rdbname.schema.table

where, for example,

Rdbname = sample

schema = connx

table = customers_db2

4. If the Owner List field is left blank, CONNX creates the sample tables with the userid as the schema/owner/library/collection name; hence the above table is created as

Sample.db2admin.customers_db2

If the specified or default schema/owner/library/collection does not exist, the attempt to build the sample tables fails with a **schema/owner/library/collection not found** SQLState/SQLCode.

Note: This example connects to a DB2 UDB 6.1 database running on Windows. The symbolic TCP/IP address is CONNXDB2. The Build Packages check box remains unchecked, which indicates that the DB2 dynamic SQL packages have already been built in the NULLID schema. These packages must be built prior to executing the DB2 sample application.

To build CONNX for DB2 sample database tables on the target host via SNA APPC LU 6.2

These tables can be built during installation of the CONNX DB2 client, or later via the MS-DOS compatibility box command line.

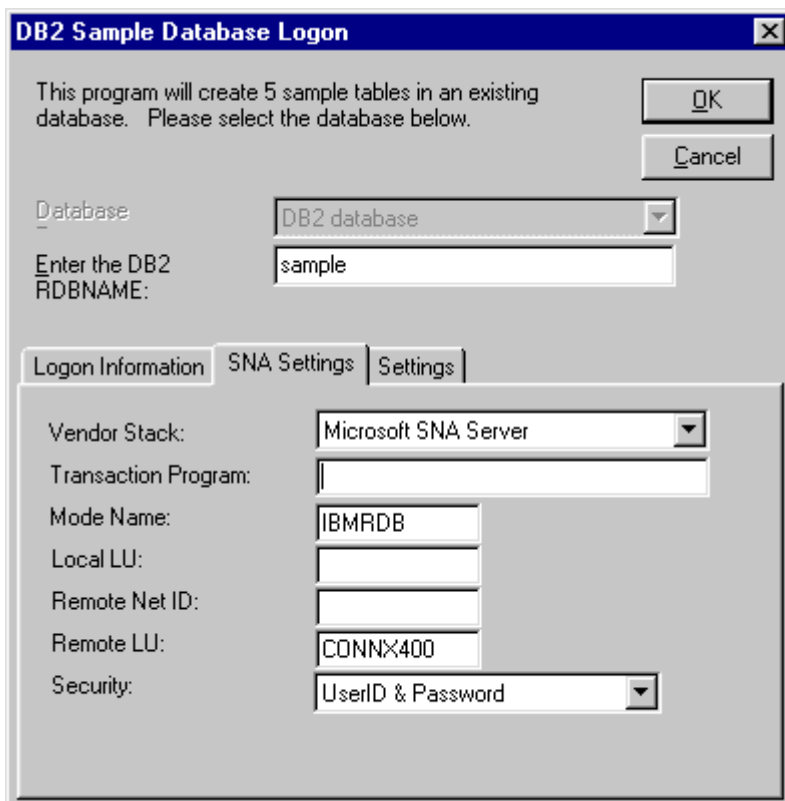
See **Step 8 - DB2 Host Configuration** for configuration information before attempting to install the sample database tables.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build DB2 Sample Data**. You may also start an MS-DOS command prompt, and then go to the CONNX installation directory. Select the **Utils** subdirectory, for example, **C:\connx32\Utils**. From the MS-DOS command prompt, start the CONNX sample tables application by typing **start sampgen db2**
2. The first **DB2 Sample Database Logon** dialog box appears after the prompt, before installation completes, or if you selected Build DB2 Sample Data in the program group.

In this example, the RDBNAME is an example; substitute the RDBNAME or location name of the target DB2 host.

3. Click the **SNA Settings** tab. The mode name is IBMRDB, which is the standard APPC mode name for DB2 access via DRDA and APPC. The Remote LU is an example; substitute values defined in your communications vendor configuration. The Vendor Stack is an example; select the SNA vendor stack installed on your client machine.

4. Click the **Settings** tab.



The owner list shows a sample schema name = CONNXDB2. All DB2 databases require three-part names for SQL objects such as tables and views. A fully qualified table name is as follows:

Rdbname.schema.table

where, for example,

Rdbname = sample
schema = connxdb2
table = customers_db2

If the Owner List is left blank, CONNX DB2 creates the sample tables with the userid as the schema/owner/library/collection name; hence the above table is created as

Sample.connx.customers_db2

If the specified schema/owner/library/collection does not exist, the attempt to build the sample tables fails with a **schema/owner/library/collection not found** SQLState/SQLCode.

5. Click the **OK** button. The DB2 sample application connects to the target host and creates five tables in the target schema/owner/library/collection. If run during CONNX installation, the DB2 sample application imports the sample table metadata into the Samples.CDD file in the \connx32\utils directory.

If you can run the DB2 sample application from the command line, you can import the created tables into any CONNX CDD. Refer to the *CONNX User Reference Guide*, which can be found online, in the CONNX product, and on the CONNX CD-ROM for a detailed explanation of import procedures.

To build Enterprise OLE DB sample database tables

The Enterprise version of the OLEDB/ODBC Adapter Module can be used to access data from any existing third-party ODBC or OLE DB data source, including IDMS, Ingres, UniVerse, Unidata, or IMS data sources. Refer to the section on CONNX OLE DB/ODBC Adapter: Enterprise and Desktop Modules in the *CONNX User Reference Guide*, which can be found online, on the CD-ROM or within the CONNX product, for more information on the text boxes required for building sample tables.

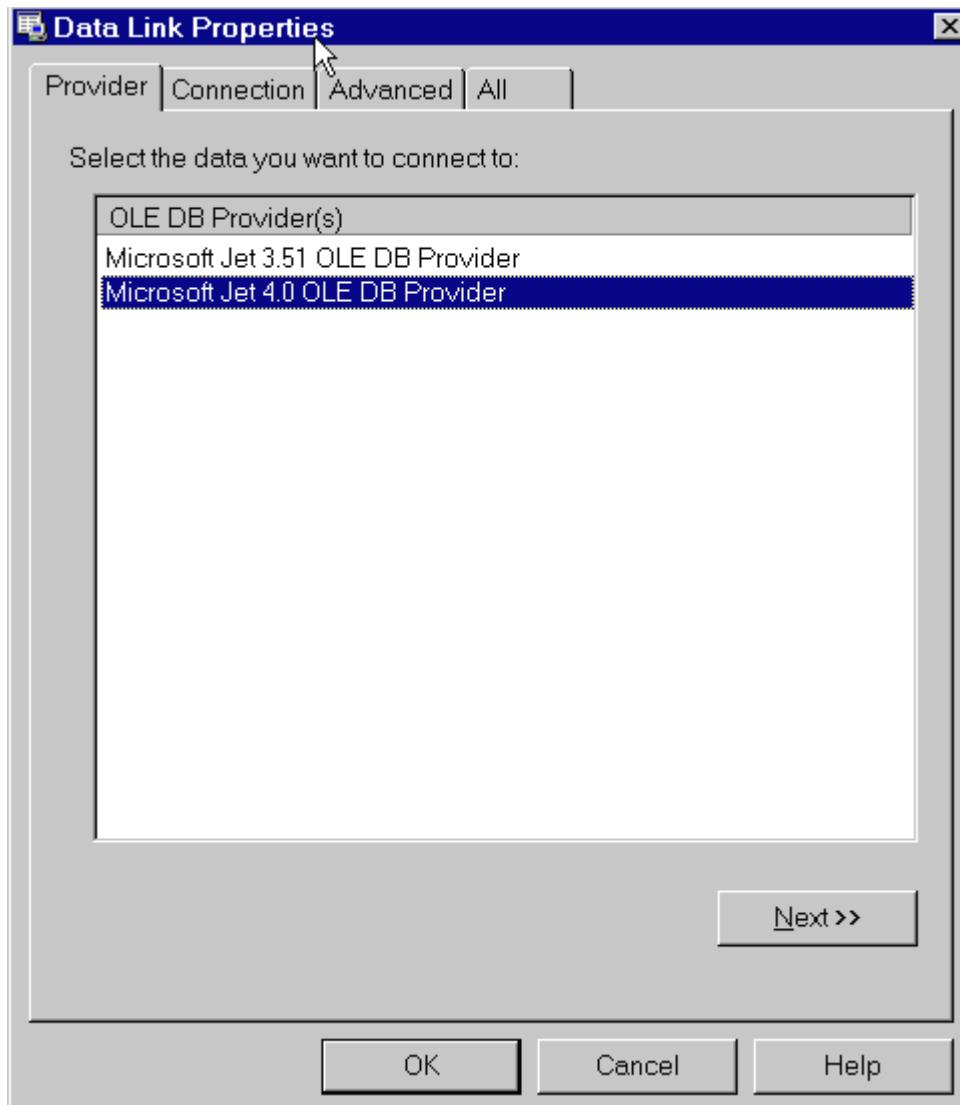
The following procedure runs a script designed to insert CONNX sample data into an OLE DB database provider installed on the client machine. For information on how to install sample data into an ODBC database, see [To Build Enterprise ODBC Sample Database Tables](#).

Sample database tables can be created either during installation or after installation. During installation, a dialog box appears asking whether you want to install sample database tables. Click **Yes** to continue with sample table installation through CONNX sample database logon dialog boxes. Click **No** to continue with CONNX installation. You can install sample tables at any time using the Build Sample Database icons available in your program group.

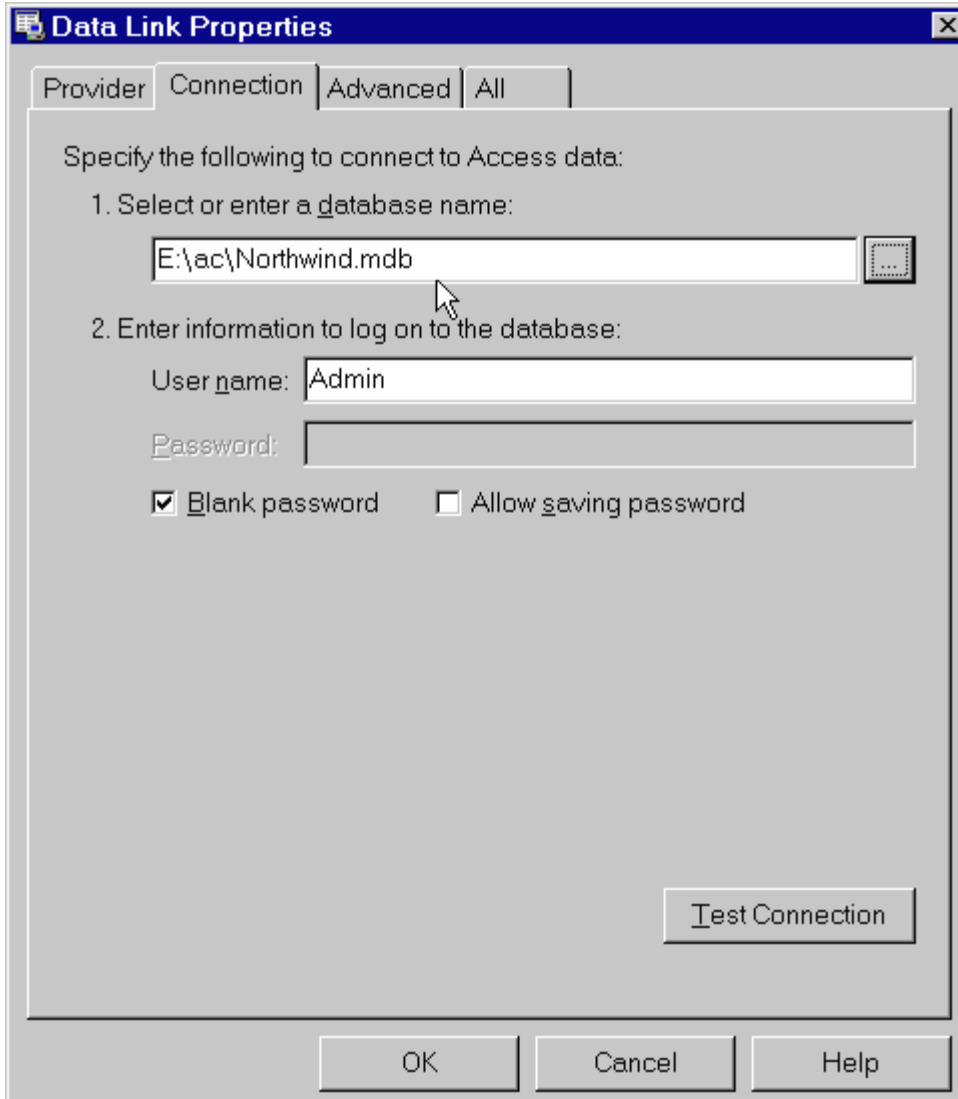
Important: The driver used to import table information must support ADO and be fully ODBC Level 2 compliant.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build Enterprise OLEDB Sample Data**. The **Enterprise OLEDB Sample Database Logon** dialog box appears. Select the **OLEDB Provider** option under **Logon Information**.

2. Click the **Select Provider** button.
3. The **Data Link Properties** dialog box appears. Select an OLE DB provider from the list. In this example, Microsoft® Jet 4.0 is used.



4. Select the **Connection** tab. Each OLE DB provider may have a unique dialog on this tab. All will require a data source or database name and possibly user and password entries. Consult documentation for each provider for specific access requirements.

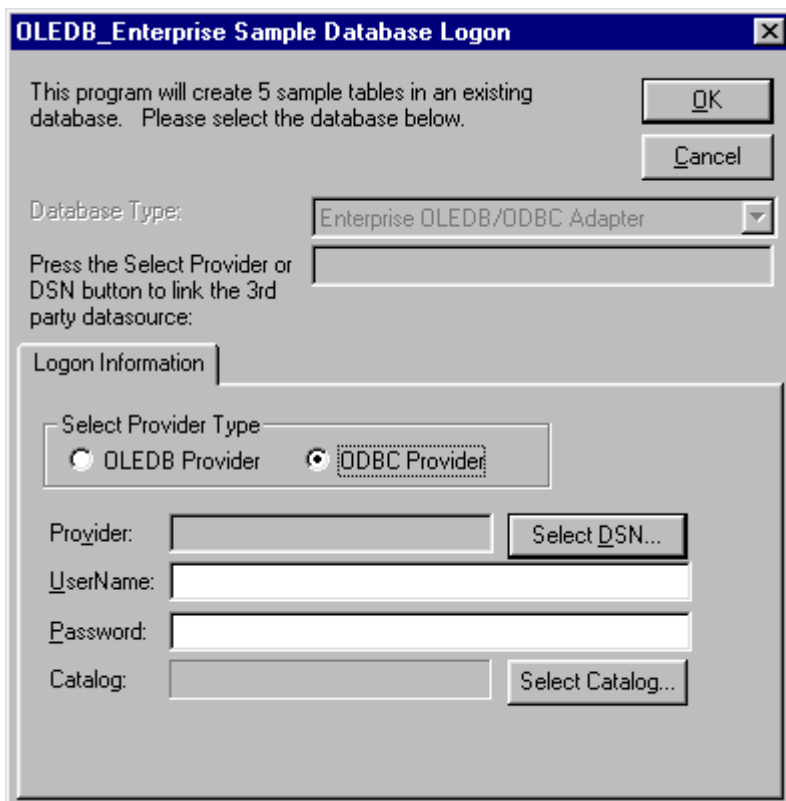


5. Click the **Test Connection** button to verify that the data source is available. Click the **OK** button.
6. The **Enterprise OLE DB Sample Database Logon** dialog box appears. Type a user name and password for the OLE DB provider, and then click the **OK** button.
7. The samples are installed in the selected data source table. You can refer to the sample tables in the post-installation stage, or by opening the CONNX Data Dictionary Manager window.

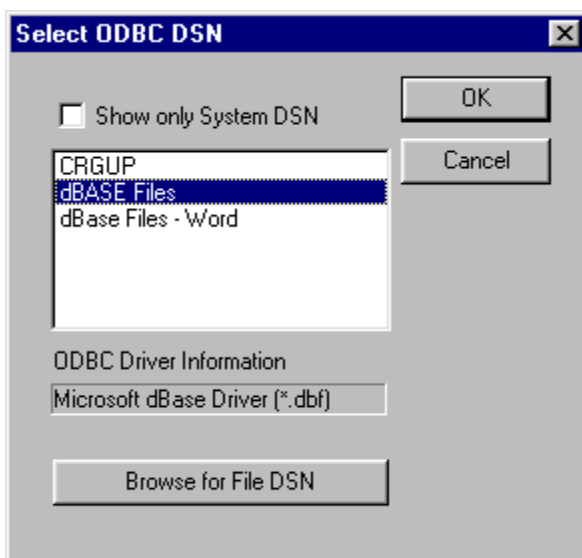
Note: This script does not delete existing tables. However, it may not work well if the selected catalog already has tables named "customers_oledb," "orders_oledb," "products_oledb," and so on. Select a catalog that does not contain such tables or drop the tables using an appropriate enterprise manager tool before proceeding with sample database import.

To build Enterprise ODBC sample database tables

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build Enterprise OLEDB Sample Data**. The **Enterprise OLEDB Sample Database Logon** dialog box appears. Select the **ODBC Provider** option under **Logon Information**.



2. Click the **Select DSN** button. The **Select ODBC DSN** dialog box appears. Select a DSN for the table in which the sample database is to be entered.



Step 2 - Install CONNX Administrator and CONNX Client

3. Click the **OK** button. The **Enterprise OLEDB Sample Database Logon** dialog box appears. Type a user name and password for that OLE DB provider.
4. Click the **OK** button to import the sample tables into the selected data source table. You can refer to the sample tables in the post-installation stage by opening the CONNX Data Dictionary Manager window.

To build Desktop OLE DB sample database tables

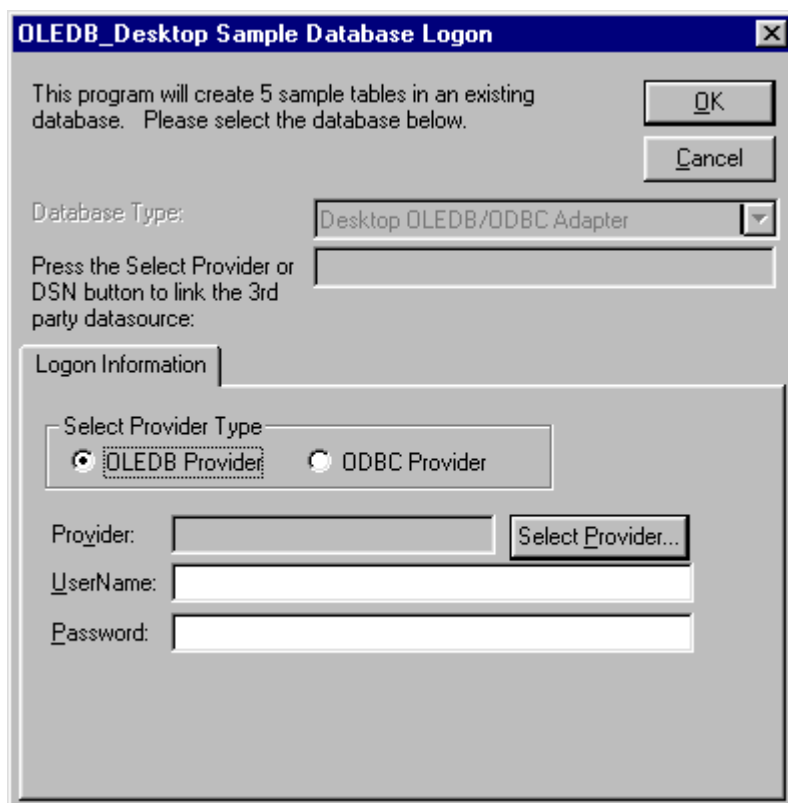
The Desktop version of the OLEDB/ODBC Adapter can be used to provide read/write access to data sources normally stored on the desktop, in a hard drive, or on a network drive, including Paradox, FileMaker, FoxPro, Lotus Notes, dbase, and Microsoft Access in database format. Refer to the *CONNX User Reference Guide*, either online, on the CD-ROM, or in hard copy, for more information on the text boxes required for building sample tables.

The following procedure runs a script designed to insert CONNX sample data into an OLE DB database provider installed on the client machine. For information on how to install sample data into an ODBC database, see [To Build Desktop ODBC Sample Database Tables](#).

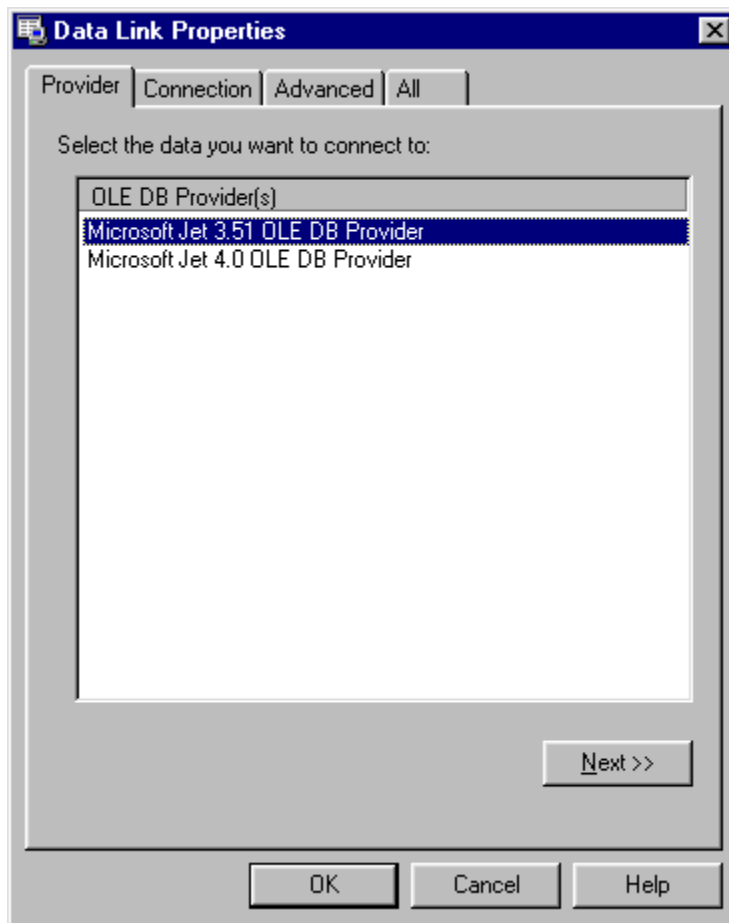
Sample database tables can be created either during installation or after installation. During installation, a dialog box appears asking whether you want to install sample database tables. Click **Yes** to continue with sample table installation through CONNX sample database logon dialog boxes. Click **No** to continue with CONNX installation. You can install sample tables at any time using the Build Sample Database icons available in your program group.

Important: The driver used to import table information must support ADO and be fully ODBC Level 2 compliant.

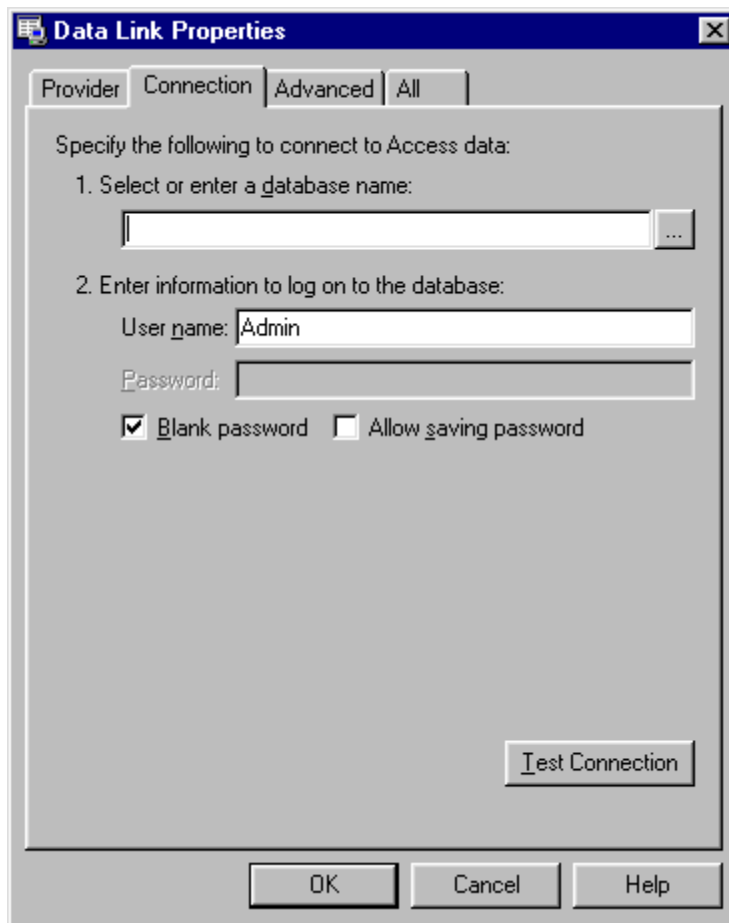
1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build Desktop OLEDB Sample Data**. The **Desktop OLEDB Sample Database Logon** dialog box appears. Select the **OLEDB Provider** option under **Logon Information**.



2. Click the **Select Provider** button.
3. The **Data Link Properties** dialog box appears. Select an OLE DB provider from the list. In this example, Microsoft Access (Jet 3.51) provider is used.



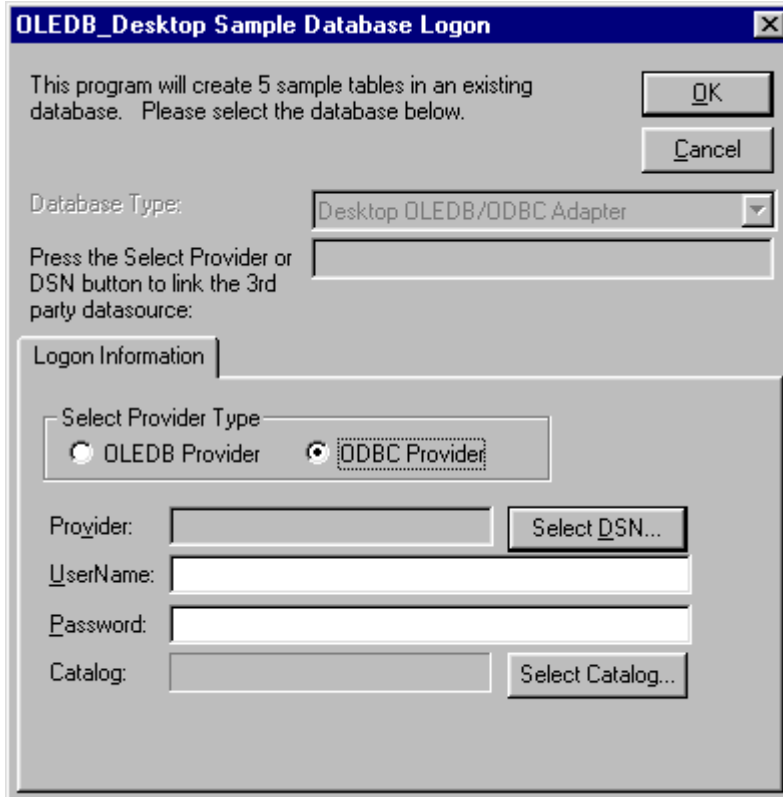
4. Click the **Next** button or select the **Connection** tab. Each OLE DB provider may have a unique dialog on this tab. All require a data source or database name and possibly user name and password entries. Consult documentation for each provider for specific access requirements.
5. Enter an OLE DB data source or click the **Browse** button (...) to locate a suitable data source for insertion of the sample data.



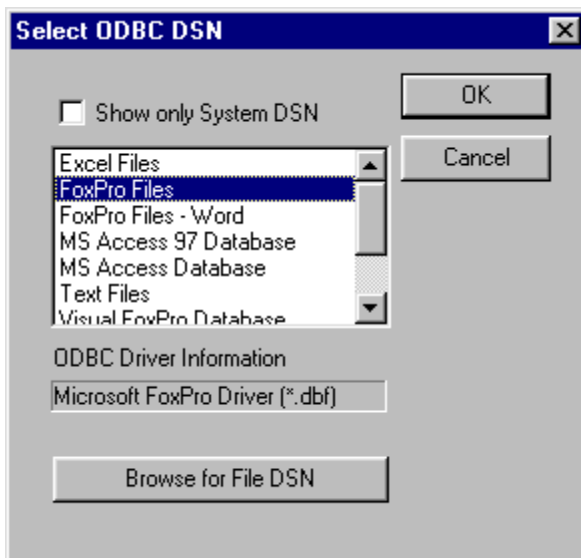
6. Click the **Test Connection** button to verify that the data source is available. Click the **OK** button.
7. The Data Link Properties window appears. Click the **OK** button to return to the **OLEDB Desktop Sample Database Logon** dialog box.
8. The **Desktop OLE DB Sample Database Logon** dialog box appears. Type a user name and password for the OLE DB provider, and then click the **OK** button.
9. The samples are installed in the selected data source table and appear in the CONNX Data Dictionary Manager window.

To install Desktop ODBC sample database tables

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build Desktop OLEDB Sample Data**. Select the **ODBC Provider** option under **Logon Information** in the **Desktop OLEDB Sample Database Logon** dialog box.



2. Click the **Select DSN** button. The **Select ODBC DSN** dialog box appears. Select a DSN for the table in which the sample database is to be entered.



- Click the **OK** button. The **Desktop OLEDB Sample Database Logon** dialog box appears. Type a user name and password for the ODBC provider.

OLEDB_Desktop Sample Database Logon

This program will create 5 sample tables in an existing database. Please select the database below.

OK Cancel

Database Type: Desktop OLEDB/ODBC Adapter

Press the Select Provider or DSN button to link the 3rd party datasource: FoxPro Files;MSDASQL.1;

Logon Information

Select Provider Type

☐ OLEDB Provider ☒ ODBC Provider

Provider: MSDASQL.1 Select DSN...

UserName:

Password:

Catalog: Select Catalog...

- Click the **OK** button. The samples are installed in the selected data source table and appear in the CONNX Data Dictionary Manager window.

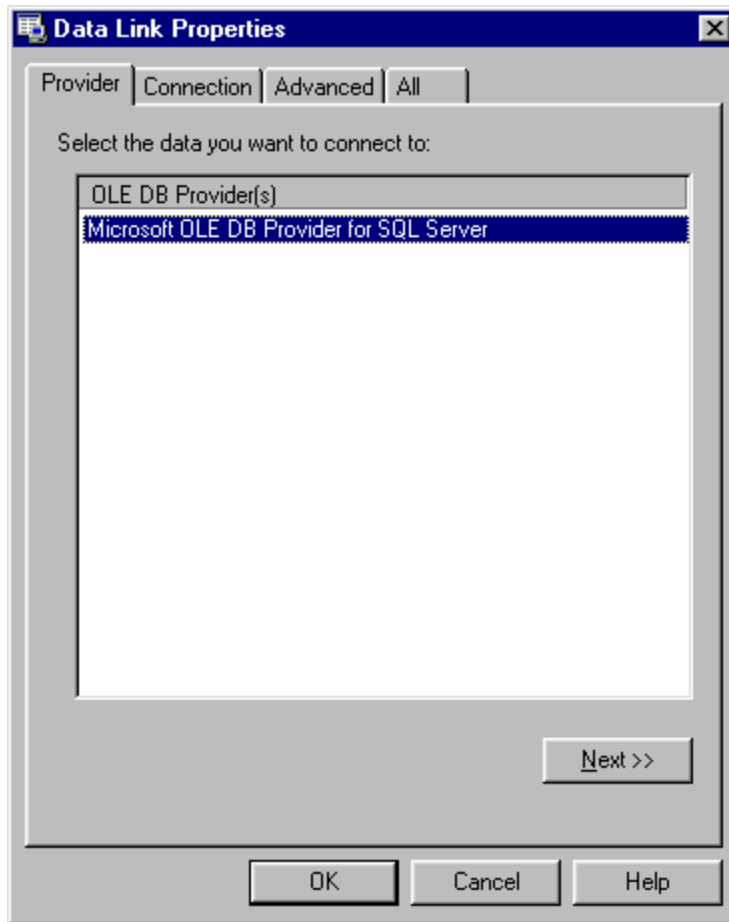
To build SQL Server OLE DB sample database tables

The CONNX SQL Server module enables access to SQL Server database objects through the Microsoft SQL Server driver. CONNX has been tested with SQL Server 7.0 and SQL Server 2000. Refer to the section on the CONNX SQL Server Database Module in the *CONNX User Reference Guide*, which is available online, within the CONNX product, and on the CONNX CD-ROM, for more information on the text boxes required to import SQL database tables.

Note: SQL Server 7.0 can use OLE DB or ODBC provider connections.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build SQL Server Sample Data**. The **SQL Server Sample Database Logon** dialog box appears. Select the **OLEDB Provider** option under **Logon Information**.

2. Click the **Select Provider** button.
3. The **Data Link Properties** dialog box appears. Select a **SQL Server OLE DB provider** from the list.



4. Click the **Next** button, or select the **Connection** tab, and then type a name of a SQL Server 7.0 host in which the sample database tables are to be installed. Type a user name and password for the SQL Server OLE DB data source provider in item 2.

The screenshot shows the 'Data Link Properties' dialog box with the 'Advanced' tab selected. The dialog has four tabs: 'Provider', 'Connection', 'Advanced', and 'All'. The 'Advanced' tab contains the following fields and options:

- Section: Specify the following to connect to SQL Server data:
 - 1. Select or enter a server name: A text box with a dropdown arrow and a 'Refresh' button.
 - 2. Enter information to log on to the server:
 - ☐ Use Windows NT Integrated security
 - ☒ Use a specific user name and password:
 - User name: [Text box]
 - Password: [Text box]
 - ☐ Blank password ☐ Allow saving password
 - 3. ☒ Select the database on the server: [Dropdown menu]
 - ☐ Attach a database file as a database name:
 - [Text box]
 - Using the filename: [Text box] [Browse button]
- [Test Connection button]
- [OK] [Cancel] [Help buttons]

5. Select a database/catalog in item 3. If Item 3 is left blank, the default catalog for the given user is used.
6. Click the **Test Connection** button to verify that the data source is available. Click the **OK** button.
7. The Data Link Properties dialog box appears. Click the **OK** button to return to the SQL Server Sample Database Logon dialog box.
8. The **SQL Server Sample Database Logon** dialog box appears. Type a user name and password for CONNX, and then click the **OK** button.
9. The samples are installed in the selected SQL Server data source table. You can refer to the sample tables in the post-installation stage by opening the CONNX Data Dictionary Manager window.

To build SQL Server ODBC sample database tables

For multiple client connections to SQL Server via ODBC through a single CONNX Data Dictionary, either the SQL Server data source name must be a file-based DSN and installed on a shared network server, or the same DSN must be configured on each client computer accessing the sample database tables.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and click **Build SQL Server Sample Data**. The **SQL Server Sample Database Logon** dialog box appears. Select the **ODBC Provider** option under **Logon Information**.

SQL Server Sample Database Logon

This program will create 5 sample tables in an existing database. Please select the database below.

Database Type: SQL Server

Press the Select Provider or DSN button to link the 3rd party datasource:

Logon Information

Select Provider Type

☐ OLEDB Provider ☒ ODBC Provider

Provider: Select DSN...

UserName:

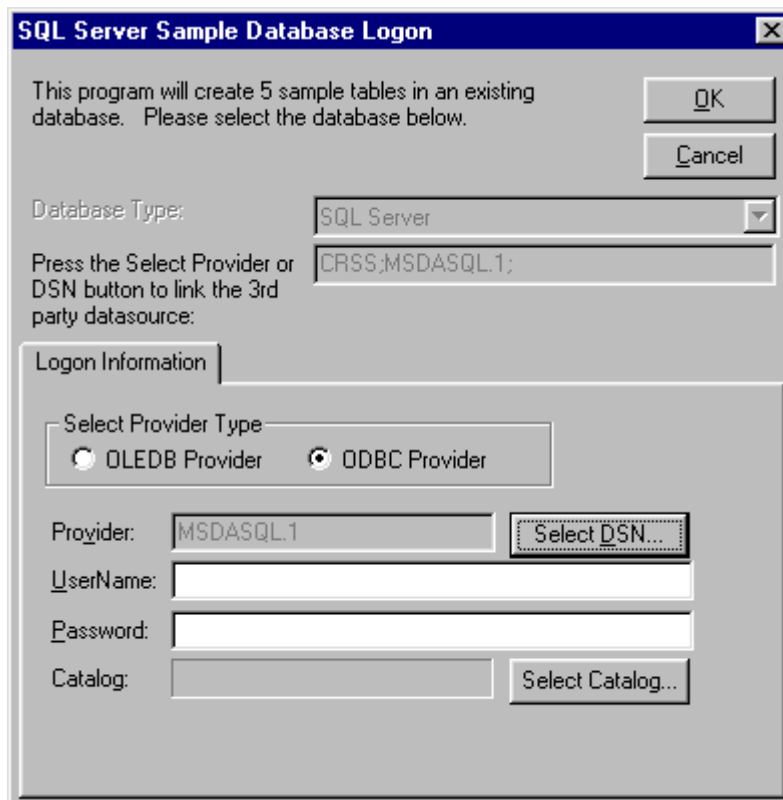
Password:

Catalog: Select Catalog...

2. Click the **Select DSN** button. The **Select ODBC DSN** dialog box appears. Select a DSN for the table in which the sample database is to be entered.



3. Click the **OK** button. The **SQL Server Sample Database Logon** dialog box appears. Type a user name and password for the ODBC connection.



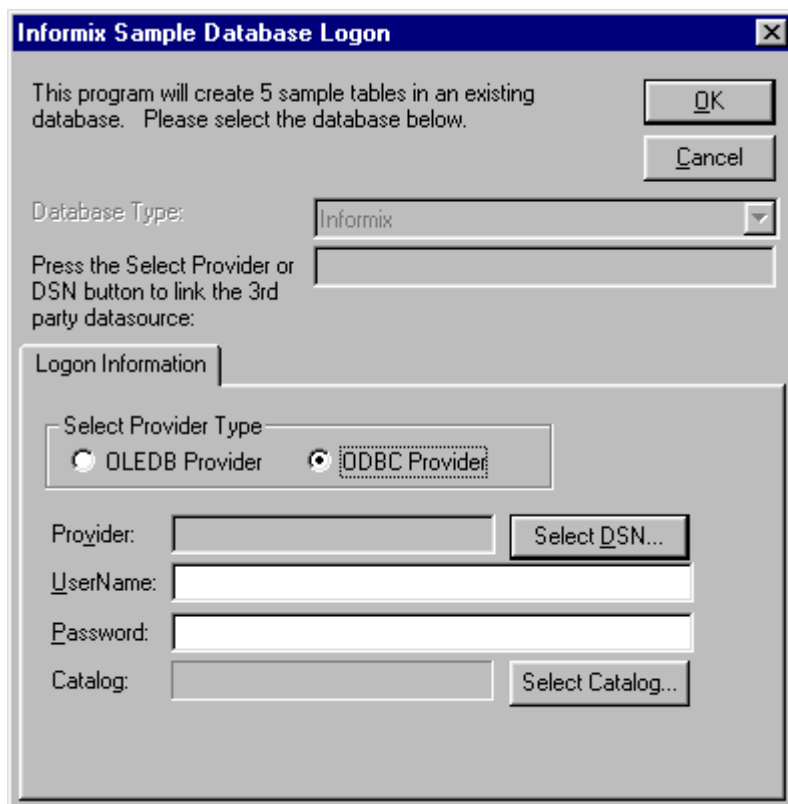
4. Click the **OK** button. The sample tables are installed in the selected data source and appear in the CONNX Data Dictionary Manager window.

To build Desktop Informix sample database tables

The CONNX Desktop Informix module enables access to Informix database objects through the Informix driver. See the section on the CONNX Informix Module in the *CONNX User Reference Guide*, either online, on the CD-ROM, or in hard copy, for more information on the text boxes required to import Informix database tables.

For multiple client connections to Informix via ODBC through a single CONNX Data Dictionary, either the SQL Server data source name must be a file-based DSN and installed on a shared network server, or the same DSN must be configured on each client computer accessing the sample database tables.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and then click **Build Desktop Informix Sample Data**. Select the **ODBC Provider** option under **Logon Information** in the **Informix Sample Database Logon** dialog box.



2. Click the **Select DSN** button. The **Select ODBC DSN** dialog box appears. Select a DSN for the data source in which the sample tables are to be entered.



3. Click the **OK** button. The **Informix Sample Database Logon** dialog box appears. Type a user name and password for the ODBC provider.
4. Click the **OK** button. The samples are installed in the selected data source table and appear in the CONNX Data Dictionary Manager window.

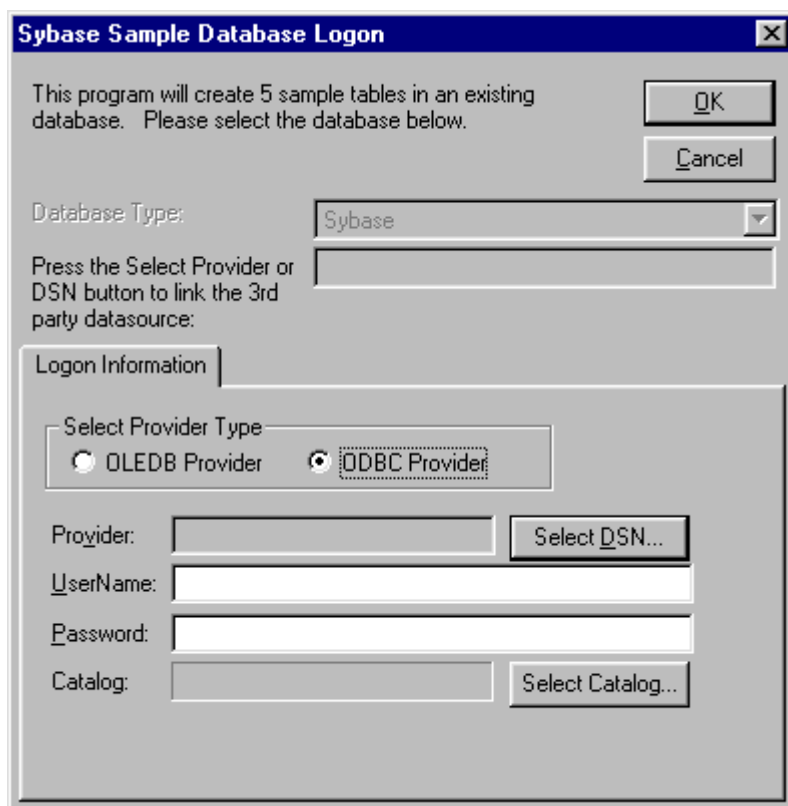
Note: Informix catalogs are not supported by the CONNX Desktop Informix module.

To build Desktop Sybase sample database tables

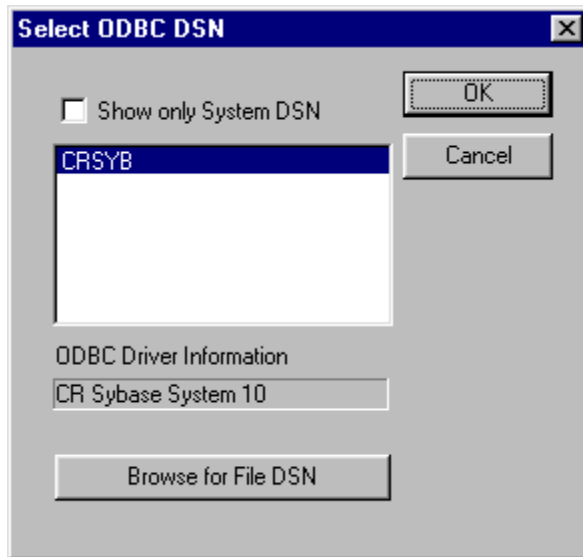
The CONNX Desktop Sybase module enables access to Informix database objects through the Sybase driver. See the section on the CONNX and Sybase Module in the *CONNX User Reference Guide*, which is available [online](#), on the CONNX CD-ROM, and within the CONNX product, for more information on the text boxes required to import Sybase database tables.

For multiple client connections to Sybase via ODBC through a single CONNX Data Dictionary, either the SQL Server data source name must be a file-based DSN and installed on a shared network server, or the same DSN must be configured on each client computer accessing the sample database tables.

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and then click **Build Sybase Sample Data**. Select the **ODBC Provider** option under **Logon Information** in the **Sybase Sample Database Logon** dialog box.



2. Click the **Select DSN** button. The **Select ODBC DSN** dialog box appears. Select a DSN for the data source in which the sample tables are to be entered.



3. Click the **OK** button. The **Sybase Sample Database Logon** dialog box appears. Type a user name and password for the ODBC provider.
4. Click the **OK** button. The samples are installed in the selected data source table and appear in the CONNX Data Dictionary Manager window.

Note: Sybase catalogs are not supported by the CONNX Desktop Sybase module.

To build C-ISAM sample database tables

The CONNX Desktop C-ISAM module enables access to C-ISAM database objects through the C-ISAM driver. See the section on the CONNX and C-ISAM Module in the *CONNX User Reference Guide*, which is available online, on the CD-ROM, and within the CONNX product, for more information on the text boxes required to import C-ISAM database tables.

Sample database tables can be created either during installation or after installation. During installation, a dialog box appears asking whether you want to install sample database tables. Click **Yes** to continue with sample table installation through CONNX sample database logon dialog boxes. Click **No** to continue with CONNX installation. You can install sample tables at any time using the Build Sample Database icons available in your program group.

The following procedure runs a script designed to insert CONNX sample data into a C-ISAM database installed on a C-ISAM server or, in Windows, on a local host machine. See [General Information for C-ISAM for configuration information](#).

1. During installation, click **Yes** to install sample database tables, or click the **Start** button, and then point to **All Programs**. Point to **CONNX Sample Databases**, and then click **Build C-ISAM Sample Data**. The **C-ISAM Sample Database Logon** dialog box appears.

2. Enter the TCP/IP host name or address for your system platform, a privileged user account name, and a user account password in the login text boxes, for example, **hpux**, **NickD**, and **Password**. (The password appears as *********.) Note that these are all required fields.

Important: If you are running C-ISAM on a Windows platform, enter "localhost" in the **Server** text box.

3. Port 6500 is listed in the **TCPIP Port** text box by default. Any change made to the port setting in this text box becomes a permanent change to the port setting of the imported database. See [Editing the OpenVMS Site-Specific Startup Command](#) for information about changing the port setting on the server.
4. Select a **Destination Database** in the list box. See "Add a Database Connection" in the *CONNX User Reference Guide*, available online, on the CONNX CD-ROM, and within the CONNX product, for more information.
5. Click the **OK** button. The samples are installed in the selected data source table and appear in the CONNX Data Dictionary Manager window.

Step 3 - Install CONNX JDBC Server and Client

CONNX JDBC General Information

If you are just getting started with CONNX JDBC, the first step you must take is to install or verify the installation of a current Java Development Kit (JDK).

To verify the previous installation of a JDK, type `java` on the command line of your application (Windows, Macintosh, Unix, Linux). If you receive no response or an error message, you either do not have a JDK installed or the JDK you have is installed incorrectly. You must download a JDK compatible with your platform. See JDBC Resources for a list of resources describing JDK downloads.

After installing or verifying the installation of a JDK, there are a few steps you need to take to complete the installation procedure, depending on your system, as follows:

- If you are working on a Windows machine, follow the procedure described in [Starting and Stopping the CONNX JDBC Server on Windows Systems](#).
- If you are working on a non-Windows machine, transfer the .jar file. See [Moving a Client to a Non-Windows Platform by Transferring the .jar File](#) for instructions.
- If you are using a non-Windows Web server, such as HotJava, Apache, or Netscape Navigator, start the CONNX JDBC Router. See [Activating the CONNX JDBC Router](#) for instructions.

For further explanation of the options available to users of CONNX JDBC, refer to the section on the CONNX JDBC Driver in the *CONNX User Reference Guide*.

Important: Before starting to use CONNX JDBC you must have Data Source Names (DSN) set up for each of the databases you intend to use. See the CONNX JDBC Driver section in the CONNX User Reference Guide for information on how to set up a JDBC DSN.

Important Note for Linux/Unix client installations: Prior to installing the Linux/Unix client, CONNX must be installed on the Windows PC where the client installer is run from. Failure to do this will result in installation errors when installing the Linux/Unix client components.

Related Topics

 [JDBC Resources](#)

 [Connecting to CONNX JDBC Server](#)

CONNX JDBC Server

The CONNX JDBC server component is a Windows executable that communicates with the CONNX JDBC driver. When the CONNX JDBC server accepts a new connection, it creates a new thread that is dedicated towards communicating to that client. There must be at least one CONNX JDBC Server running on either a LAN or WAN for CONNX JDBC clients to operate.

Related Topics

 [Starting and Stopping the CONNX JDBC Server on Windows Systems](#)

 [Connecting to CONNX JDBC Server](#)

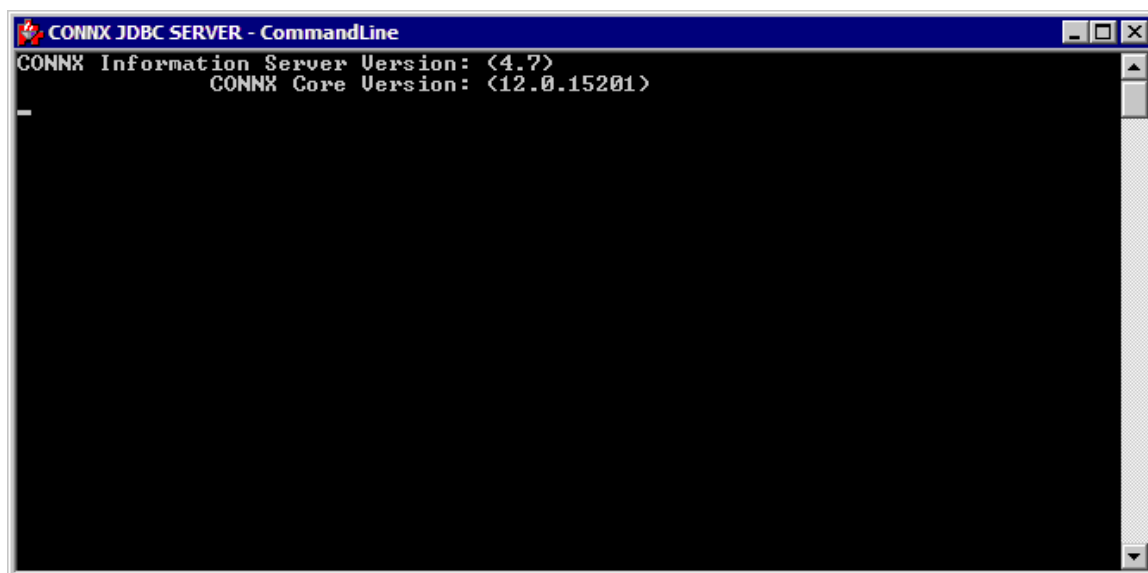
Starting and Stopping the CONNX JDBC Server on Windows Systems

The CONNX JDBC server runs as a service on Windows systems and this step is not necessary.

If you wish to run the CONNX JDBC server from the command line rather than as a service, you must first stop the CONNX JDBC Server Service and then follow the steps outlined below.

To start the CONNX JDBC Server from the command line on Windows systems

1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX JDBC Server (Command Line)**.
2. The CONNXJDBC window appears and minimizes to the task bar.



Note: During installation of the CONNX JDBC Server, a dialog box appears asking for user name and password if the CONNX JDBC service needs to be registered to a specific account. The local system account usually does not have rights to network drives, and since the CONNX JDBC Service needs access to the network (because of license files), the service must be bound to a user who has access rights.

To stop the CONNX JDBC Server on Windows systems

There are three ways to stop the CONNX JDBC server, as follows:

1. Maximize the CONNXJDBC window, and then click the **Close** button.
2. Right-click on the **CONNXJDBC** icon on the taskbar, and then click **Close**.
3. Select the **CONNXJDBC** icon, and then click **Control <C>** within the window.

Activating the CONNX JDBC Router

If either of the following is true in your environment, you need to deploy the CONNX JDBC Router.

- You want your client machines to be able to execute applets that use JDBC on a non-Windows server.
- Your Internet and intranet clients cannot connect to port 7500 on the machine running the CONNX JDBC server because of firewall issues.

If either of the above statements is true, the CONNX JDBC Router must be deployed. The CONNX JDBC Router is a Java executable that acts as if it were the CONNX JDBC Server. Clients connect to the IP address of the machine on which the router is located (instead of the CONNX JDBC Server). The router then accepts client requests, but instead of processing the requests, dispatches the packet to the CONNX JDBC Server. The router then receives a reply from the CONNX JDBC Server, and sends it back to the client. This dispatcher function enables clients to function with firewalls or preexisting applet security.

If both of the statements are not true, this step is unnecessary. Proceed to [Connecting to the CONNX JDBC Server](#) to continue.

For more information on the CONNX JDBC Router, see CONNX JDBC Router in the *CONNX User Reference Guide*, available either online, on the CONNX CD-ROM, or within the CONNX product.

To activate the CONNX JDBC Router

1. Install the content of the router on a non-Windows Web server. It will be necessary to have Classpath defined to point to the correct JDK. The Classpath should include the connxrouter.jar file.

Example:

```
export CLASSPATH=<unix installation
directory>/java/ftp/connxrouter.jar;:$CLASSPATH
```

For more information about setting the Classpath, refer to CONNX JDBC Driver, in the *CONNX User Reference Guide*, either online, on your CONNX CD-ROM, or within the CONNX product. For more information about transferring the .jar file, see [Moving a Client to a Non-Windows Platform by Transferring the .jar File](#).

2. Invoke the Router by typing the following command on the command line of your native machine.

```
java com.Connrx.Router.TCRouter <ip address or name of CONNX JDBC
Server> <Server Port Number> <Router Port Number>
```

The following table contains a description of the above syntax:

Router syntax

Syntax	Description
IP Address	IP address of your machine or host name of your machine.
Server Port Name	Port number from 0 to 99. Default is 7500.
Router Port Number	Port number created by user that does not conflict with exiting Port number.

Related Topics

[Moving a Client to a Non-Windows Platform by Transferring the .jar File](#)

Connecting to the CONNX JDBC Server

There are several steps that must be followed in order to connect to the CONNX JDBC server.

1. Make sure the database is registered using the **DSNRegistry tool**, which is described in [CONNX JDBC Driver](#) in the *CONNX User Reference Guide*, available [online](#), on the CONNX CD-ROM, or within the CONNX product.
2. Set **Classpath**, which is described in "CONNX JDBC Driver" in the online *CONNX User Reference Guide*.
3. Load the **JDBC Driver**, which is described below.
4. Set the **CONNX JDBC URL**, which is described in "CONNX JDBC Driver" in the *CONNX User Reference Guide*.
5. Open a connection to a database via JDBC.

To load the JDBC Driver

- To load and register the CONNX JDBC driver, invoke the method `Class.forName` in JDBC, supplying the classname as follows:
`Class.forName("com.Connex.jdbc.TCJdbc.TCJdbcDriver").newInstance();`

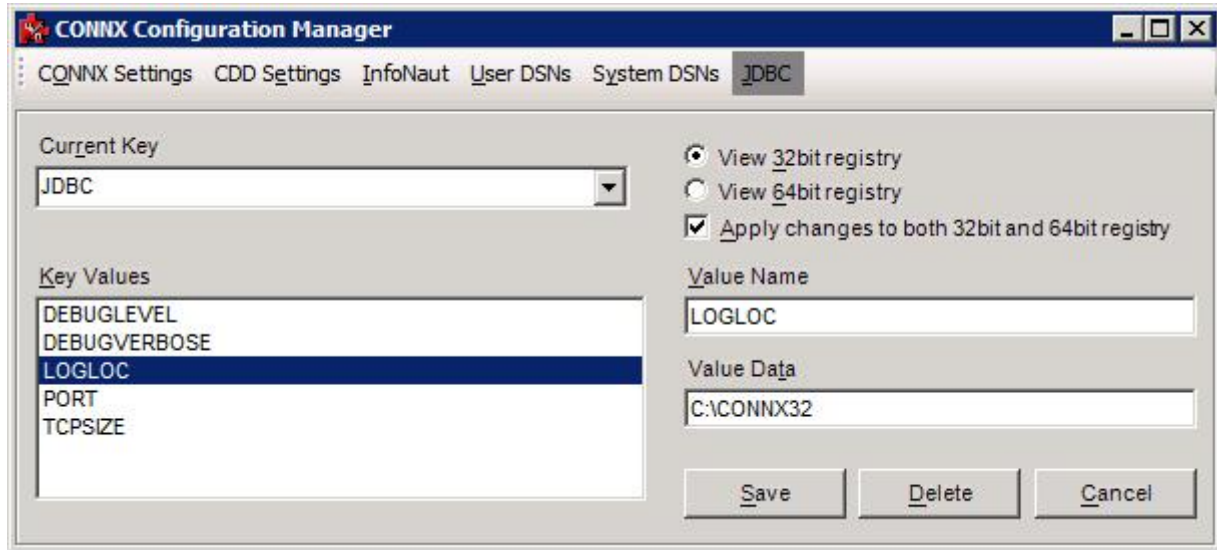
Related Topics

 [Activating the CONNX JDBC Router](#)

 [Moving a Client to a Non-Windows Platform by Transferring the .jar File](#)

CONNX JDBC Configuration Settings

The Configuration Manager can adjust some of the CONNX JDBC configuration settings to increase the flexibility of the CONNX JDBC Server.



Note: This only applies in a Windows setting.

Note: This screenshot shows the 32bit registry value which pertains to the 32bit JDBC server. To change settings for the 64bit JDBC server, please check the radio button labeled "View 64bit registry".

For more information about the using the Configuration Manager to adjust the CONNX JDBC configuration settings, see the CONNX Configuration Settings chapter in the User Reference Guide.

The CONNX JDBC Server on Unix

Starting and Stopping the CONNX JDBC Server on UNIX

These steps are required after installation and before running any database connections for UNIX systems. See the online help describing connection procedures for your specific data source.

To start the CONNX JDBC Server on UNIX systems

1. Go to the CONNX directory that was created in the installation directory specified during installation of the CONNX UNIX Client. The cnxjdbc executable file resides in that folder.
2. Verify that your data source names (DSNs) are registered correctly.

At the command prompt, enter **./sqlregistry**

Select option **1** to view all the current registry settings

If your DSN is in the list, go to the next step

If your DSN is not in the list, enter **CONNX.DSNS.dsn that you want to register**

Select option **2** for a string value

Enter the fully qualified path to the CONNX CDD. The CDD must be located on a UNIX system.

Add a semicolon. [optional] Add a DSN comment freeform string. Add a semicolon.

Example:

```
/opt/connx/MyData.cdd;;
```

Enter 5 to exit the the /sqlregistry program

Alternatively you may utilize the createJDBCDSN utility script to manage your JDBC DSNs. Information about the createJDBCDSN utility script is located at the end of this section.

3. At the UNIX terminal session prompt, enter the server command to run the executable as a background process:
./jdbcserver START

This will create a connxjdbc.log file that contains JDBC Server information, error messages, and status.

```

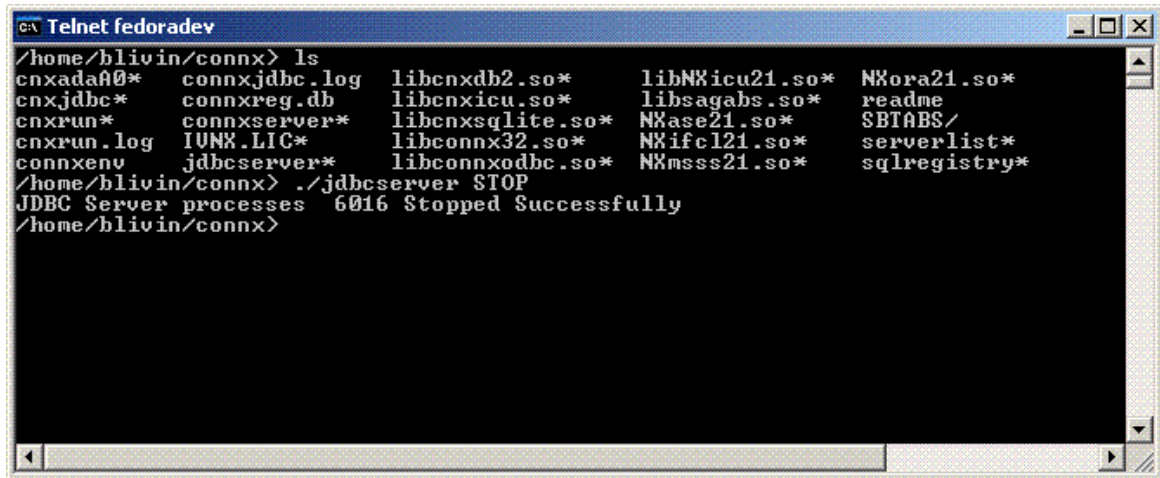
C:\ Telnet fedoradev
/home/blivin/connx> ls
cnxada00*  connxjdbc.log  libcnxdb2.so*  libNXicu21.so*  NXora21.so*
cnxjdbc*  connxreg.db    libcnxicu.so*  libsagabs.so*  readme
cnxrun*   connxserver*  libcnxsqlite.so*  NXase21.so*  SBTABS/
cnxrun.log IUNIX.LIC*   libconnx32.so*  NXifcl21.so*  serverlist*
connxenv  jdbcserver*  libconnxodbc.so*  NXmsss21.so*  sqlregistry*
/home/blivin/connx> ./jdbcserver START
JDBC Server: 6016 /HOME/BLIVIN/CONNX/CONNXJDBC
JDBC Server started successfully
/home/blivin/connx> _

```

To stop the CONNX JDBC Server on UNIX systems

To stop the JDBC Server, at the UNIX terminal session prompt, enter the server command to run the executable as a background process:

```
./jdbcsrvr STOP
```



```

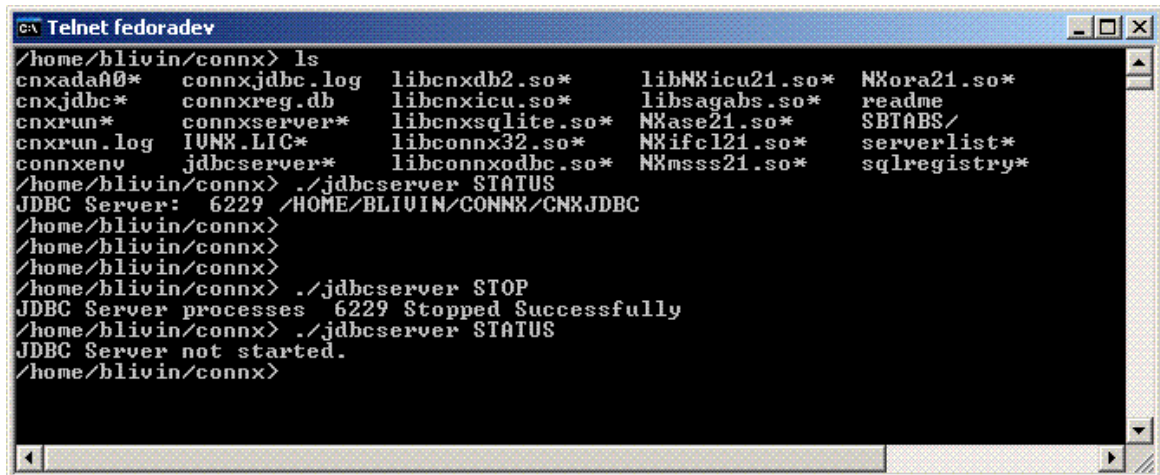
Telnet fedoradev
/home/blivin/connx> ls
cnxada00*  connxjdbc.log  libcnxdb2.so*  libNXicu21.so*  NXora21.so*
cnxjdbc*  connxreg.db    libcnxicu.so*  libsagabs.so*  readme
cnxrun*   connxserver*   libcnxsqlite.so*  NXase21.so*    SBTABS/
cnxrun.log  IUNIX.LIC*    libconnx32.so*  NXifc121.so*   serverlist*
connxenv   jdbcsrvr*     libconnxodbc.so*  NXmsss21.so*   sqlregistry*
/home/blivin/connx> ./jdbcsrvr STOP
JDBC Server processes 6016 Stopped Successfully
/home/blivin/connx>

```

To verify that the CONNX JDBC Server on UNIX systems is active

To verify that the JDBC Server is active, at the UNIX terminal session prompt, enter the server command to run the executable as a background process:

```
./jdbcsrvr STATUS
```



```

Telnet fedoradev
/home/blivin/connx> ls
cnxada00*  connxjdbc.log  libcnxdb2.so*  libNXicu21.so*  NXora21.so*
cnxjdbc*  connxreg.db    libcnxicu.so*  libsagabs.so*  readme
cnxrun*   connxserver*   libcnxsqlite.so*  NXase21.so*    SBTABS/
cnxrun.log  IUNIX.LIC*    libconnx32.so*  NXifc121.so*   serverlist*
connxenv   jdbcsrvr*     libconnxodbc.so*  NXmsss21.so*   sqlregistry*
/home/blivin/connx> ./jdbcsrvr STATUS
JDBC Server: 6229 /HOME/BLIVIN/CONN/CONN/JDBC
/home/blivin/connx>
/home/blivin/connx>
/home/blivin/connx>
/home/blivin/connx> ./jdbcsrvr STOP
JDBC Server processes 6229 Stopped Successfully
/home/blivin/connx> ./jdbcsrvr STATUS
JDBC Server not started.
/home/blivin/connx>

```

Using the createJDBCDSN utility script

```
./createJDBCDSN -l
```

This form of the createJDBCDSN script will list all configured JDBC DSNs.

```
./createJDBCDSN <DSN> <CDD> [comment]
```

DSN: Top level name of CONNX JDBC DSN

CDD: Fully qualified path to CONNX CDD

comment: [optional] comment

This form of the createJDBCDSN script will add a new DSN to the registry or will overwrite an existing DSN entry. If a DSN entry already exists the user will be prompted before overwriting the old value.

To edit the JDBC server startup procedure in UNIX

1. Log on or use telnet to connect to your UNIX based JDBC server and start a terminal session.
2. Use your favorite UNIX editor (Emacs, for example) to create a shell script from the following text:

```
# begin CONNX startup script
CNXDIR=<path to CONNX JDBC server>
PATH=<basic system path for shell programs>
export CNXDIR PATH
case "$1" in
  start)
    su <connx user> -c "$CNXDIR/jdbcserver START";;
  stop)
    su <connx user> -c "$CNXDIR/jdbcserver STOP";;
  restart|reload)
    su <connx user> -c "$CNXDIR/jdbcserver START";;
  esac
exit 0
#end CONNX startup script
```

3. Supply the correct values for <path to CONNX JDBC server>, <basic system path for shell programs> and <connx user>. The value for <path to CONNX JDBC server> will be the path to the CONNX installation directory. The value for <basic system path for shell programs> should contain the path for basic shell utilities such as cd and ls. The value for <connx user> will be the user that has been designated to start the CONNX processes. It is not recommended that any of the CONNX server processes be started with the root account.
4. Save the document in the appropriate UNIX flavor. The following table contains some example of file names:

UNIX Flavor	File Name
HP-UX	/sbin/rc2.d/S999connx
LINUX	/etc/rc.d/rc5.d/S99connx
SCO	/etc/rc2.d/S99connx
SUN	/sbin/rc2.d/S99connx
SOLARIS	/sbin/rc2.d/S99connx
TRU-64	/sbin/rc2.d/S99connx
AIX	<p>If you do not have rc#.d directories, where # is the run level number and generally 2 for multi-user, then you can put the following lines toward the bottom of the /etc/rc script. For example:</p> <pre>CNXDIR=path to CONNX JDBC server export CNXDIR sh\$CNXDIR/connxserver START [port number]</pre>

--	--

If an example of the file name to use for your UNIX flavor does not appear in the above table, contact CONNX Technical Support for details.

The CONNX JDBC server automatically starts when you restart the system. To verify that it has started, type the following UNIX command after a command prompt in your UNIX system:

```
$CNXDIR/jdbcserver STATUS
```

4. Check the log file in the CNXDIR directory to verify the start procedure.

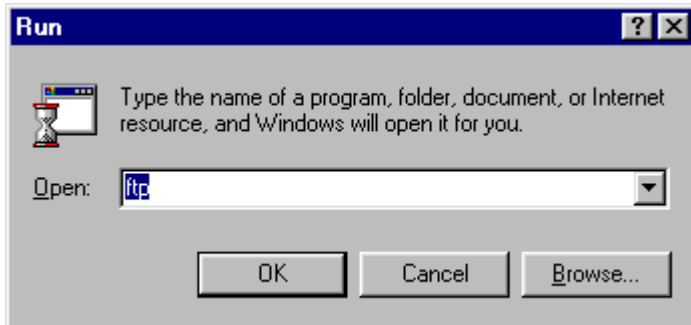
Note: *Since the startup procedures vary between operating systems as well as UNIX flavors, the above examples should be used as a rough guide. Consult your operating system's system administration guide for the exact location of startup scripts and directories. You should be logged in as root to complete the script configuration and set the proper permissions on the script.*

Moving a Client to a Non-Windows Platform By Transferring the .jar File

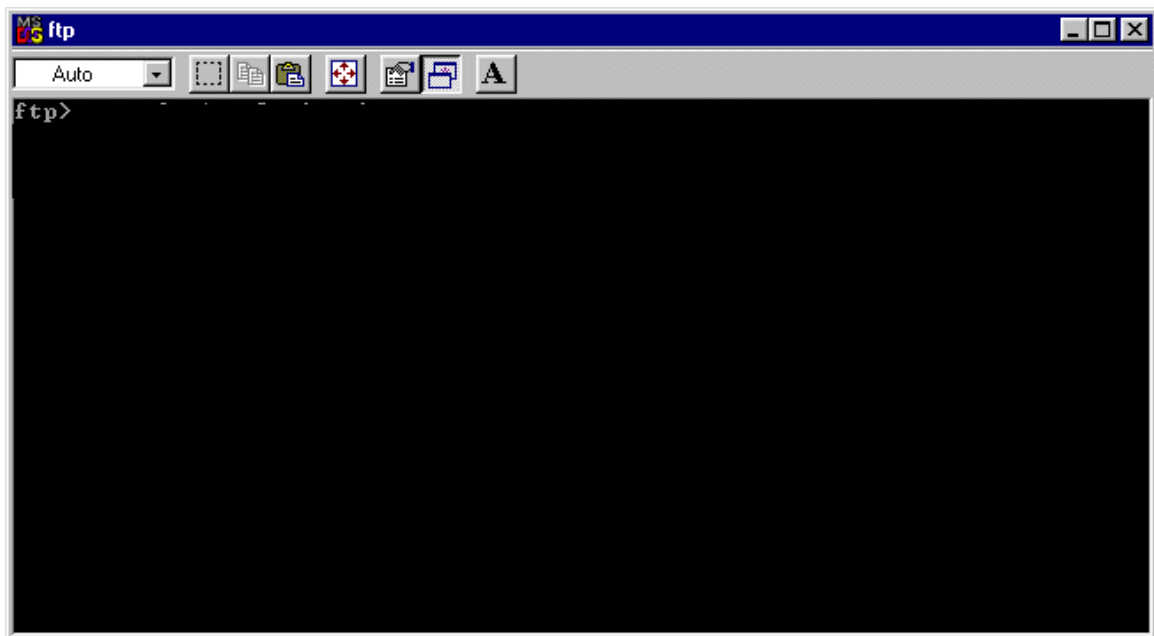
This step is not necessary if your client targets are Windows machines. However, if you are working on a non-Windows machine, you must transfer the .jar file from its installation location to a location on the non-Windows machine.

To move a client to a non-Windows platform

1. Click **Start**, point to **Programs**, and select **Windows Explorer**. Locate the Windows directory on your hard drive, for example C:\Windows or C:\Winnt. Select the **ftp.exe** file. You can also click **Start**, click **Run**, type **ftp** in the **Open** text box, and then click the **Open** button.



2. The ftp window appears.



3. Enter your user name and password.

```
ftp> open <ftpl.yourcompany.com> or <machine name> or <machine ip
address or alias>
```

```
Connected to ftpl.yourcompany.com.
220 (your machine name) Microsoft FTP Service (Version x.0).
User (ftpl.yourcompany.com:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as
password.
```

```

Password: <enter password>
230-Welcome to Your Company
230-in Computing. Please e-mail any problems to
230-ftpmaster@yourcompany.com.
230 Anonymous user logged in.

```

4. Specify binary to insure that files will be transferred as binary. Binary mode is necessary to insure that the files are transferred correctly.

```
ftp> binary
```

5. Navigate to the destination directory. A user optional subdirectory should be selected on the target machine. This is where the CONNX JDBC client files reside. In the example below, the user selects /users/njit as the transfer destination.

```

ftp> cd /users
250 CWD command successful.

ftp> cd njit
250 CWD command successful.

```

6. Send **connxjdbcftp.jar** to the destination directory. The CONNX JDBC client files are stored in an archive called **connxjdbcftp.jar** located in **C:\CONN32\CONNXJDBC\java\ftp**. The user must specify the fully qualified name of **connxjdbcftp.jar**.

```
ftp > put C:\CONN32\CONNXJDBC\
java\ftp\CONNXJDBCftp.jar
```

7. Log on to the target system and go to destination directory. Users must log on to the target system with an appropriate username and password, and should change the current directory to be the location of the **connxjdbcftp.jar** file.

```

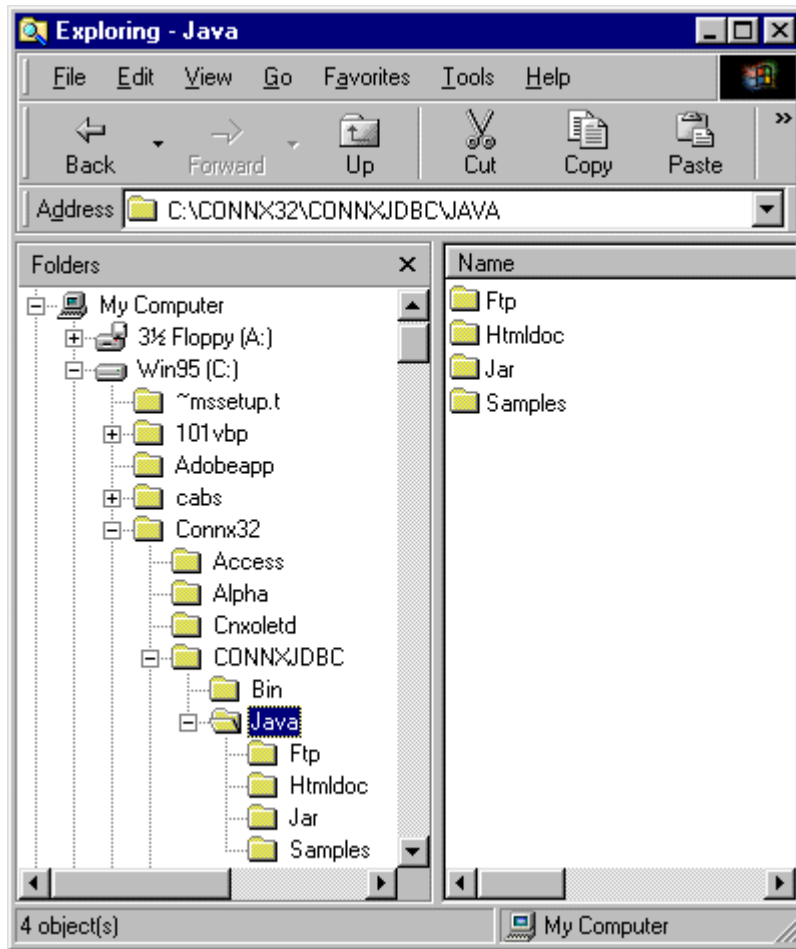
Welcome to <your Computer>
Username: <your User ID>
Password: <your Password>

```

8. Unzip the **connxjdbcftp.jar** file. The user must run the jar program in order to extract the files from **connxjdbcftp.jar**.

```
jar xvf connxjdbcftp.jar
```

9. The files are extracted and a directory structure is built. The subdirectory is java; below that is Ftp, HtmlDoc, Jar, and Samples, following the same structure of the files installed on your Windows or Winnt/System32 directory.



Related Topics

- >> [Starting and Stopping the CONNX JDBC Server on Windows Systems](#)
- >> [Activating the CONNX JDBC Router](#)

Step 4 - Install CONNX OpenVMS Server








General Information Regarding the OpenVMS Server

This step copies the OpenVMS components from a computer to an OpenVMS server. The CONNX Administrator component must be installed on the computer.

If you are installing Oracle, DB2, VSAM, C-ISAM, DISAM, Micro Focus, Adabas, or OLE DB/ODBC providers, skip this step and continue to [Step 9 - Post-Installation Checklist](#).

After the installation of the components on the OpenVMS server, run an installation program on the server to set environment variables, set up files, and build the CONNX server function.

Related Topics

-  [About the OpenVMS User Account and Password](#)
-  [CONNX OpenVMS Installation Procedure](#)
-  [Testing OpenVMS Server Installation](#)
-  [Post-Installation Checklist - OpenVMS Server](#)
-  [Editing the OpenVMS Site-Specific Startup Command](#)
-  [Reviewing the Minimum OpenVMS Quotas Procedure](#)
-  [Testing a Database Connection](#)

About the OpenVMS User Account and Password

When installing the OpenVMS components:

Use a user account name and password with privileges that allow you to create subdirectories and to copy, read, and write files to the server. It is recommended that your OpenVMS system administrator conduct this procedure.

When using the CONNX login to connect to data:

Both the OpenVMS user name and password are stored in the CONNX CDD. If the OpenVMS user name changes on the OpenVMS computer system, the CONNX user name and password must be authenticated through the CDD.

Related Topics

 [CONNX OpenVMS Installation Procedure](#)

CONNX OpenVMS Installation Procedure

Use the following procedure to install the CONNX OpenVMS server components. Before starting this procedure, be sure to review [About the OpenVMS User Account and Password](#).

To install the CONNX OpenVMS components on an OpenVMS server

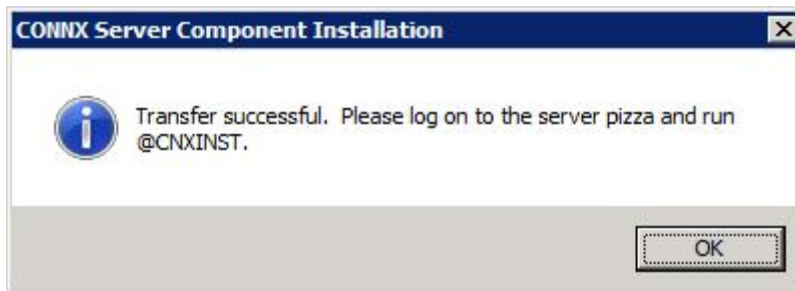
1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the VMS tab on the CONNX Server Component Installation dialog box.

2. In the **Installation Location** frame, select whether this will be a shared or individual location.

Note: The Shared Location option can only be selected for an account that has WRITE and CREATE privileges in the SYS\$COMMON directory.

- **Individual Location Account.** Select this option for a single-user installation of CONNX. When this option is selected, the following occur on the OpenVMS server:
 - Directory Machine1:[NickD,CONNX] is created, where Machine 1 is the server drive and NickD is the User Account name, and CONNX server installation files are copied from the administrator computer to this directory
 - Directory SYS\$COMMON:[CONNX] is created and CONNX server installation files are copied from the administrator computer to this directory
 - CONNX files are copied from the administrator computer to SYS\$LIBRARY

- **Shared Location Account.** Select this option for CONNX multi-user and demo installations. When this option is selected, the following occur on the OpenVMS server:
 - Directory SYS\$COMMON:[,CONNX] is created and CONNX server installation files are copied from the administrator computer to this directory
 - CONNX files are copied from the administrator computer to SYS\$LIBRARY
- 3. **Login Information.** Type the VMS Server Name for the OpenVMS server in the **Server** text box, a privileged user account name in the **User ID** text box, and its password in the **Password** text box.
- 4. **Product.** Select the database(s) for which you have licenses.
- 5. **Server Type.** Select the server type on which your OpenVMS system resides.
- 7. Click the **Install** button. Allow several minutes for the file transfer process between the CONNX administrator computer and the OpenVMS server. After the transfer process is complete, the **Installation Successful** dialog box appears.



- 8. Click the **OK** button. The **Installation Successful** dialog box closes. The CONNX server files have been successfully copied to the OpenVMS server. Click the **Close** button.
- 9. On the CONNX administrator computer, start a terminal emulator and log into the account used to copy files to the server. The \$ prompt appears.
- 10. After the \$ prompt, type **@cnxinst.com** and then press **<Enter>**. The monitor displays installation results from the **cnxinst.com** command. For example, if you are installing Rdb, a successful installation displays:

CONNX for Rdb has installed successfully!

If there are problems, an unsuccessful installation displays:

The CONNX Server setup did NOT complete successfully.

Note: If the installation fails for an Individual Location account, you have insufficient privileges to perform these actions. In this case, continue with the Individual Account installation and, when finished, have your OpenVMS system administrator perform the License File Only procedure.

Note: If the procedure fails for a Shared Location account, it means the user account has insufficient access privileges. Either change the access privileges for this user account or select a user account with sufficient privileges.

- 11. After the **cnxinst.com** execution is complete (allow several minutes), the \$ prompt is returned. If the installation was unsuccessful, you may have used a user name with insufficient privileges. Choose another user name and try the installation again.
- 12. Review the **CNXSETUP.LOG** file to analyze the installation results. To do this, after the \$ prompt, type **edit/edt CNXSETUP.LOG** and press **<Enter>**.
 - After the * character, type **c** and then press **<Enter>** for a full window display. Use the arrow keys to scroll through the file

- Look for a string of *** characters that appear around the installation messages; for example, if the installation was successful, look for

```
$ write cnx$screen "*****"
```

```
$ write sys$output "CONNX for RMS has installed successfully!"
```

- Or, if the installation was not successful, look for:

```
$ write cnx$screen "*****"
```

```
$ write sys$output "The CONNX Server setup did NOT complete successfully."
```

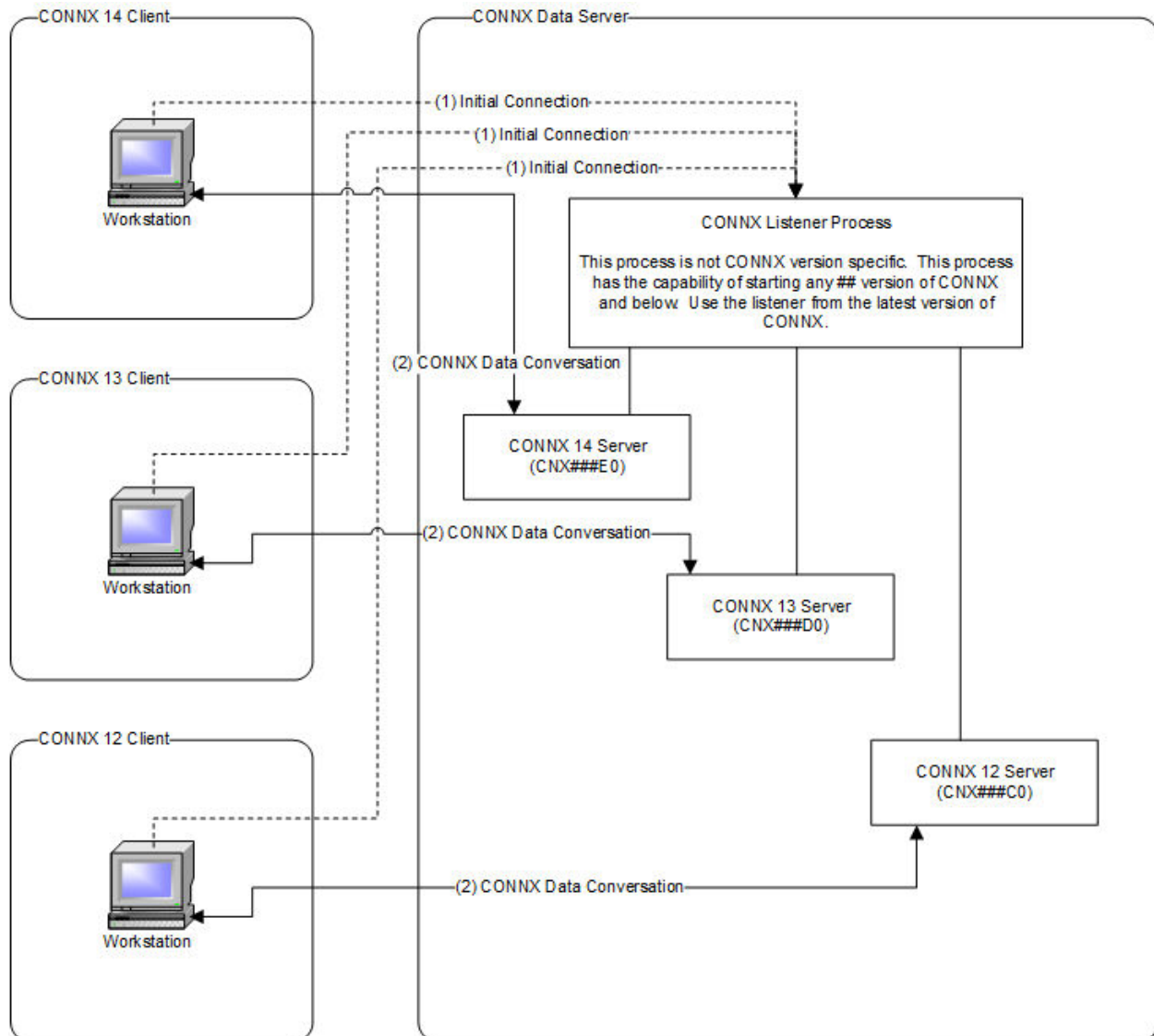
Note: You can transfer the CNXSETUP.LOG to your computer for analysis. See [Reviewing CNXSETUP.LOG Using FTP -- TCP/IP](#).

13. Disconnect and close the terminal emulator.
14. Continue by testing the OpenVMS server installation. See [Testing the OpenVMS Server Installation](#).

Tip: If an error message appears, check your entries for OpenVMS server name, user account name, and password, then try again.

Running Multiple Versions of CONNX on an OpenVMS Server

Multiple versions of CONNX can be run on an OpenVMS server platform. When installing a new version of CONNX, install the CONNX Client, CONNX Administrator, OpenVMS Server Components, along with the Help and Samples files on the Windows client machine. When the installation is nearing completion, you are prompted to install the CONNX OpenVMS component on an OpenVMS server.



CONNX connections are made in two steps:

The initial connection is made from the CONNX client to the CONNX listener. CNXRUN##_MAIN is the CONNX listener process. In this step, the CONNX listener starts up the appropriate version of the CONNX server, matching the CONNX client version to the CONNX server version.

All subsequent communication occurs directly between the newly started CONNX process and the CONNX client.

Note: The CNXRUN##_MAIN listener is only able to start up older client versions if the logicals and .exe files of the appropriate versions are available.

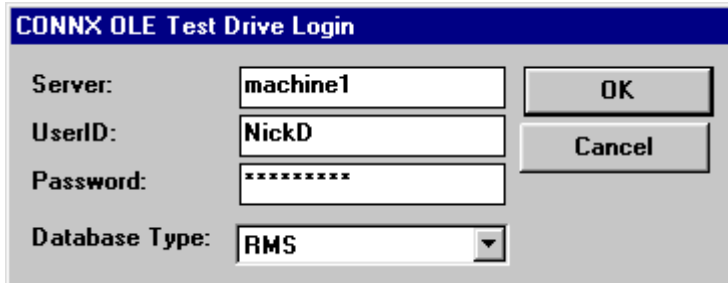
Related Topics

 [CONNX Installation Procedure](#)

 [CONNX OpenVMS Installation Procedure](#)

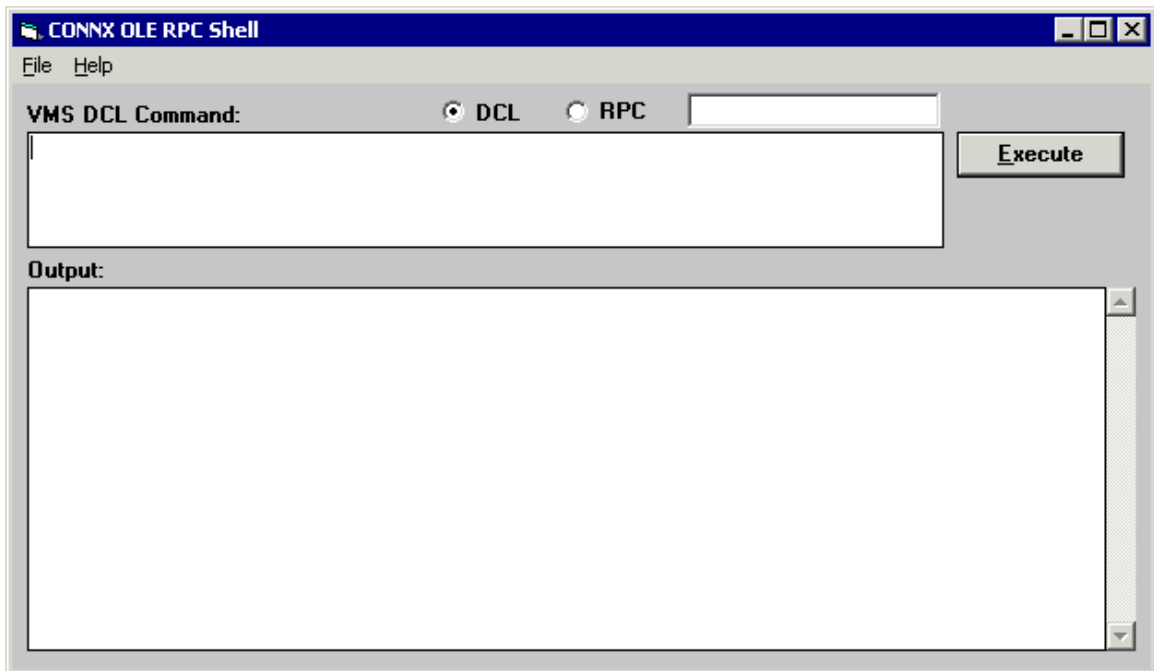
Testing the OpenVMS Server Installation

1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **Visual Basic VMS OLE Example**. The **CONNX OLE Test Drive Login** dialog box appears.



The dialog box titled "CONNX OLE Test Drive Login" has a blue title bar. It contains four input fields: "Server:" with the text "machine1", "UserID:" with the text "NickD", "Password:" with masked characters "*****", and "Database Type:" with a dropdown menu showing "RMS". To the right of the input fields are two buttons: "OK" and "Cancel".

2. Enter a server, user name, password, and then select a database type in the list box; for example, **Machine1**, **NickD**, **Password**, and **RMS**. Then click the **OK** button. The CONNX OLE RPC Shell window appears.



The window titled "CONNX OLE RPC Shell" has a blue title bar and a menu bar with "File" and "Help". Below the menu bar is a section with "VMS DCL Command:" followed by two radio buttons, "DCL" (selected) and "RPC", and a text input field. To the right of this section is an "Execute" button. Below the command section is a large text area labeled "Output:".

3. Type **dir** and then click the **Execute** button. The results of the directory command are displayed in the **Output** pane.
4. This completes the CONNX OpenVMS Installation Procedure.

Testing User Account Privileges

This procedure tests the privilege level of an OpenVMS user account by creating and then deleting a SYS\$COMMON:[CNXTESTDIR] test directory. If successful, this user account may be used for installing CONNX.

Important: Request permission from your system administrator before running this test because it creates and deletes a directory on the OpenVMS server.

To test the privileges of an OpenVMS user account

1. On a CONNX administrator computer, start a terminal emulator, select the desired server, and type the appropriate user name and password. The \$ prompt appears.
2. Follow this abbreviated sequence of OpenVMS commands:

1.	\$ create/directory SYS\$COMMON:[CNXTESTDIR]
2.	\$ set def SYS\$COMMON:[CNXTESTDIR]
3.	\$ dir
4.	\$ no files found
5.	\$ cd [-]
6.	\$ del CNXTESTDIR.DIR;1
7.	\$ SYS\$COMMON:[000000]CNXTESTDIR.DIR;1 deleted

Description:

1.	Create the directory SYS\$COMMON:[CNXTESTDIR].
2.	Set default directory to SYS\$COMMON:[CNXTESTDIR].
3.	Display contents of directory SYS\$COMMON:[CNXTESTDIR].
4.	Return "no files found" because directory is empty. This user account MAY BE USED for CONNX installation. You have sufficient privileges to create directories. Return "error opening SYS\$COMMON:[CNXTESTDIR]" indicates that you do not have sufficient privileges to create directories and therefore this user account MAY NOT BE USED for installing CONNX.
5.	Move up one directory level in SYS\$COMMON.
6.	Delete the empty directory CNXTESTDIR in SYS\$COMMON.
7.	Return indicates directory was deleted successfully.

3. Disconnect the session and close your terminal emulator.

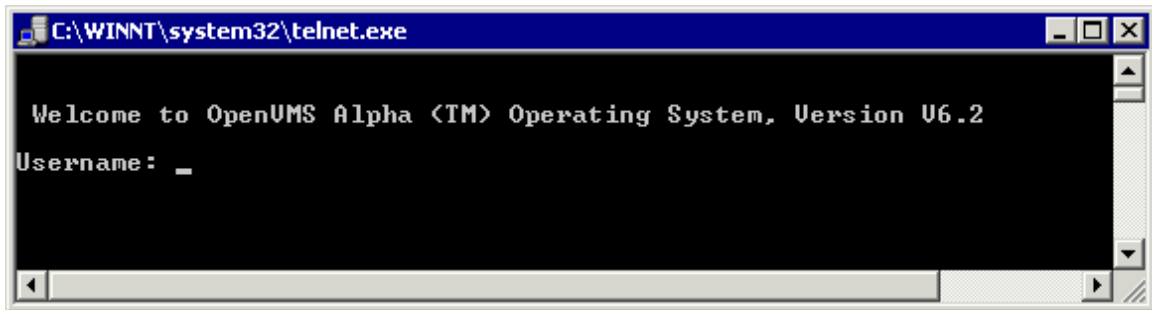
Note: Use a terminal emulator most familiar to you or see [Using Your Telnet Utility - TCP/IP](#).

Using Your Telnet Utility - TCP/IP

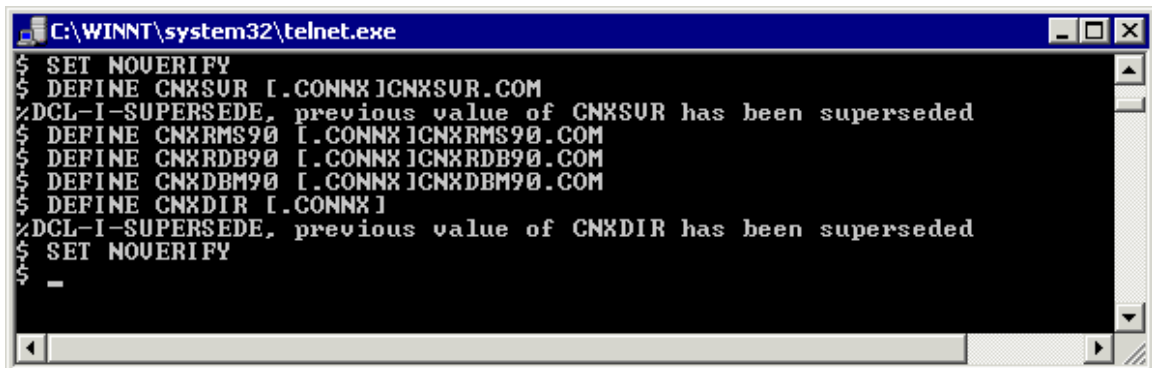
When your network protocol is TCP/IP, you can use the Telnet terminal emulator supplied with Windows to communicate with the OpenVMS server from the CONNX administrator computer.

To use Telnet to communicate with the OpenVMS server

1. Click the **Start** button, and then point to **Run**. In the **Open** text box, type **telnet <host name>** and then click the **OK** button. The `x:\WINDOWS\CONNX32\telnet.exe` window appears.



2. Enter a user name and password for a user with sufficient creating, reading, and writing privileges for your OpenVMS activity. The **Welcome to OpenVMS** message appears. Enter a user name and password for a user with sufficient creation, read and write privileges for your OpenVMS activity.
3. With the displayed prompt displayed, continue to the desired OpenVMS activity.



Note: If the welcome message does not appear, make sure your network connection to the server is working. See [Troubleshooting Connection Problems](#).

Step 5 - Install IBM Mainframe-Compatible Server

General Information

CONNX for CICS/VSAM for OS/390 / z/OS and VSE is a CICS C++ application built with the SAS/C 7.00 Cross-Compiler for Windows, a product of the SAS Institute. CONNX for CICS/VSAM can only access VSAM files defined to and managed by the target CICS region or partition. The CONNX Data Dictionary Manager metadata import utility requires that the COBOL copybooks for the target CICS VSAM files reside in OS/390 / z/OS partitioned data sets or in VSE libraries defined with FTP read access.

CONNX for VSAM / QSAM / PDS for OS/390 / z/OS is a C++ application built with the SAS/C 7.00 Cross-Compiler for Windows. CONNX for VSAM / QSAM / PDS can access any VSAM or QSAM (partitioned- or physical sequential-) data set defined on the target system. COBOL copybooks that map the target VSAM or QSAM files must reside in a partitioned data set defined with FTP read access. VSAM files which are allocated exclusively to CICS regions can be accessed only when the regions are quiesced. For read-only queries, CONNX for VSAM / QSAM / PDS can access VSAM files defined with VSAM read share options of 3 or 4 when the files are allocated to one or more CICS regions. For update processing, the target VSAM data sets must be allocated exclusively to a CONNX for VSAM / QSAM / PDS started task or batch job.

Outline of Installation Procedures

The installation procedures for CONNX for CICS/VSAM - z/OS, CONNX for VSAM / QSAM / PDS, and CONNX for VSAM - VSE are each divided into five parts.

CONNX for CICS/VSAM - z/OS

- Part 1 describes installing the CONNX CICS/VSAM components on a z/OS server.
- Part 2 covers installation and verification of the CONNX CICS components.
- Part 3 describes the use of the CONNX Configuration Utility for CICS/VSAM.
- Part 4 describes the installation of CONNX onto a second CICS region for z/OS.
- Part 5 describes the procedure used to uninstall CONNX for CICS/VSAM - z/OS.

CONNX for VSAM / QSAM / PDS

- Part 1 describes installing the CONNX for VSAM / QSAM / PDS Started Task components on an OS/390 server.
- Part 2 describes installation verification of the CONNX for VSAM / QSAM / PDS components.
- Part 3 describes the use of the CONNX TSO Configuration Utility.
- Part 4 explains how to start multiple instances of CONNX for VSAM / QSAM / PDS.
- Part 5 describes the procedure used to uninstall the CONNX for VSAM / QSAM / PDS Started Task components.

CONNX for VSAM or Adabas - VSE

- Part 1 describes how to create a CONNX library in VSAM-managed space.
- Part 2 describes installing the CONNX VSAM components on a VSE system.
- Part 3 covers installation verification of the CONNX CICS components.
- Part 4 describes the use of the CONNX Configuration Utility.
- Part 5 describes the procedure used to uninstall CONNX for VSAM or Adabas - VSE.

z/OS

Adabas SQL Gateway (CONNX for Adabas) - z/OS

Mainframe Installation

Security

Security for CONNX and Adabas on the server side is described in "[CONNX for Adabas Host / Client Security Overview](#)".

Prerequisites for Adabas

Adabas SQL Gateway (CONNX for Adabas) requires that:

- The Adabas database product be installed.
- The target system be set up to support TCP/IP communications (OE stack).
- The TCP/IP started task must be running.
- Read/write FTP access be configured and available on the target system for at least one TSO user ID for installation only.
- OSS segment must be defined for the user ID in a security system (RACF / ACF2 / Top Secret) or it must be available by default.

CONNX for Adabas is a reentrant application which executes as a started task or a batch job. During installation, the CONNX Adabas server program must be linked with the reentrant version of the Adabas batch/TSO link routine (ADALNKR). Please see the appropriate Adabas documentation for more information on ADALNKR.

Before submitting the CONNX installation job, please check with your systems programmer or Adabas administrator to ensure that the ADALNKR load module exists in a load library which can be link-edited with the CONNX for Adabas object code.

Verifying TCP/IP Connectivity

Before you install CONNX on your Adabas server, we recommend that you verify the TCP/IP connectivity to your OS/390 z/OS system. To verify connectivity:

1. Open a Command Prompt window.
2. On the taskbar, click Start and then click Run
3. Type or select:

ping hostname

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Part 1 - Adabas SQL Gateway (CONNX for Adabas) Installation Procedures

Installation Procedures

To install the CONNX components on the target z/OS system

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS UNIX/Linux

CONNX Server Data Set

CONNX DSN HLQ CONNX.ADABAS

SMS ☐ VOLSER CONNX1 UNIT 3390

Load Library Parameters

Load Library DSN CONNX.ADABAS.LOAD

VOLSER CONNX1 UNIT 3390

ADABAS Parameters

Adabas Load Lib SAG.ADALOAD

Login Information

Server

Username

Password

Product

☒ ADABAS

☐ IMS

☐ VSAM

FTP Mode

☒ Active ☐ Passive

☐ Manual Copy

RMODE

☒ Use RMODE=SPLIT

Install Done

2. Select the OS/390 / z/OS tab in the CONNX Server Component Installation dialog box. Select the ADABAS option in the **Product** area. Select the Started Task option in the **Server Type** area.
3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your OS/390 / z/OS server in the **Server** text box, a TSO user ID in the **User ID** text box and its password in the **Password** text boxes.
4. In the **CONNX Server Data Set** area, enter the data set name of the CONNX server component location (where the Adabas SQL Gateway will be installed on the mainframe) in the **CONNX DSN HQL** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text box. If SMS is installed on the CONNX server system, select the **SMS** check box.
5. In the **Load Library Parameters** area, enter the dataset name of the link step output files in the **Load Library DSN** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in

the **UNIT** text box. Enter the dataset name of the Adabas Nuclear load library in the **Adabas Load Library** text box.

NOTE: The **Load Library DSN** name is automatically generated when you enter the data set name of the CONNX server component location in the **CONNX Server Data Set** area.

6. In the **FTP Mode area**, enter the FTP mode that is appropriate for your FTP server. If the host cannot be accessed via FTP, select Manual Copy. Manual Copy will create a local copy of the files that need to be transferred along with a text file with instructions.

7. In the **RMODE** area, checking **Using RMODE=SPLIT** will generate JCL that specifies RMODE=SPLIT during the Inkedit. This options allows most of the SAS/C runtime to run above the 16MB line reducing the amount of below the line storage consumed by the data server.

To use RMODE=SPLIT, the CONNX load library must be a PDSE. The installer can determine whether or not the load library is a PDSE on zOS 1.10, zOS 1.11 and zOS 1.12 if the PTF for PM19545 has been applied or any version of zOS above 1.12.

If the installer is able to determine that an existing CONNX load library is a PDS rather than a PDSE, it will ask for permission to rename the existing load library and then create a new one as a PDSE. If you do not want the installer to rename the existing load library, you will need to either uncheck RMODE=SPLIT or convert your existing load library to a PDSE.

If the installer is unable to determine the type of library, you will be prompted to either rename, continue (because you have verified that it is a PDSE on our own) or go back to remove the RMODE=SPLIT option.

If you are installing into a load library that is not in the DSN specified by the CONNX DSN HLQ, the installer will not provide the option of doing a rename. In this case, the installer must be able to either verify that the load library is a PDSE and if it cannot, you will need to verify it before RMODE=SPLIT will be allowed.

Note: If RMODE=SPLIT is used and the load library is not a PDSE, the install JCL will fail.

8. Click the **Install** button. This action starts an FTP session to copy the selected components from the client PC to the target host. It may take several minutes for the file transfer process to complete.

9. Click **Close**.

Required and Optional Entry Fields for Mainframe TCP/IP Server Setup

Group Box	Term	Description
Login	Adabas server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.123
	User ID	Your 1-8 character TSO logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character TSO password. All characters convert from lowercase to uppercase.
Operating System	OS/390 z/OS	
Server Type	Started Task	
CONNX Server Data	CONNX DSN HLQ	A one- or multi-part high-level qualifier which is used to create the CONNX installation sequential and partitioned

Set		data sets on the target host. In the current example, the installation prefix is CONNX.CICS.
	VOLSER	The DASD volume serial on which the CONNX.Adabas sequential and partitioned data sets are created. Optional: The DASD volume serial on which the CONNX.ADABAS sequential and partitioned data sets are created.
	UNIT	The DASD device type (3380, 3390, etc.) of the above VOLSER. Optional: The DASD divide type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
Load Library Parameters	Load Library DSN	
	VOLSER	The DASD device type (3380, 3390, etc.) of the above VOLSER. Optional.
	UNIT	The DASD device type (3380, 3390, etc.) of the above VOLSER. Optional: The DASD divide type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
RMODE	RMODE=SPLIT	Specifies whether or not the SAS/C runtime will be loaded above the 16MB line
CICS Parameters	Target RDO File	
Adabas Parameters	Adabas Load Library	The fully qualified dataset name of the Adabas Nucleus load library.
	SBCCSID	See Code Page Support on the Adabas SQL Gateway in the CONNX User Reference Guide for more information. The default SBCCSID used during installation is 37.

DSN Prefix Datasets

Data Set	Description
CONNX.ADABAS.CNTL	Installation JCL and control records partitioned data set.
CONNX.ADABAS.LOAD	CONNX load library partitioned data set for Started Task TCP/IP listener/server program.
CONNX.ADABAS.OBJECT	CONNX object code PDS default dataset name.

Part 2 - Installing the Adabas SQL Gateway (CONNX for Adabas) components

Introduction

Once the ftp successfully completes, you are ready to install the selected CONNX components. To begin this process, log on to TSO on the target system. Given the CONNX Server Data set prefix above = CONNX.Adabas, the ftp process creates a JCL/control record partitioned data set with the name and members listed in the following table:

Default Partitioned Data Set Name = CONNX.ADABAS.CNTL

Member	Purpose
INST	The CONNX for Adabas installation job built by the CONNX installation program.
REMOVE	CONNX for Adabas uninstallation job + delete steps for sample files.
REMOV01	CONNX for Adabas uninstallation job - delete steps for sample files.
CNXLINKA	JCL procedure to execute the IBM linkage editor against CONNX prelinked SAS/C 7.00 object code and create the CONNX for Adabas load modules.
CNXPARMS	Sample control records which define SAS/C environment variables for the CONNX for Adabas TCP/IP Listener and Server programs (CNXRUNA and CNXAD'X'0B) where 'X' is the version number (10=A, 11=B, 12=C, 13=D, etc).
CNXADA	JCL procedure to invoke the CONNX for Adabas TCP/IP Listener/Server programs to access the CONNX for Adabas sample files.
CNXADAJ	Execution JCL for procedure CNXADA.
LNKRUNA	Linkage editor control records for load module CNXRUNA.
LNKADAB	Linkage editor control records for load module CNXAD'X'0B, where 'X' is the version number (10=A, 11=B, 12=C, 13=D, etc).

Edit the INST member and change the job card to match your site standards; then submit this job to the batch execution queue. All job steps must end with condition codes less than or equal to 4.

VM - [24 x 80]

File Edit View Communication Actions Window Help

Menu Functions Confirm Utilities Help

VIEW CONNX.ADABAS.CNTL Row 00001 of 00014
 Command ==> Scroll ==> PAGE ID

Name	Prompt	Size	Created	Changed	ID
CNXADA					
CNXADAJ					
CNXLINKA					
CNXLINKW					
CNXPARMS					
CNXSTOP					
CNXSTOPJ					
DDRUN					
INST					
LNKADAB					
LNKRUNA					
LNKSTART					
LNKSTOP					
REMOVE					
End					

04/015

Connected to remote server/host zvm5c using port 23

Session D - [24 x 80]

File Edit View Communication Actions Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT CONNX.ADABAS.CNTL (INST) - 01.00 Columns 00001 00072
 Command ==> Scroll ==> CSR

***** Top of Data *****

```

000001 //CNXINST JOB (),
000002 //      'CONNX INSTALLATION',
000003 //      MSGCLASS=H,
000004 //      MSGLEVEL=(1,1),
000005 //      NOTIFY=&SYSUID ,
000006 //*      USER=,
000007 //*      PASSWORD=,
000008 //*      REGION=0M,
000009 //*      TIME=(2,59) ,
000010 //*
000011 // JCLLIB ORDER=CONNX.ADABAS.CNTL
000012 //*****
000013 //*** STEP 1.1: CREATE A TEMPORARY LOAD LIBRARY: ***
000014 //*****
000015 //CRTTEMP EXEC PGM=IEFBR14
000016 //CNXTEMP DD DSN=&&CNXLOAD,
000017 //      DISP=(,PASS,DELETE) ,
000018 //      UNIT=SYSDA,
000019 //      SPACE=(TRK,(5,15,1)) ,
  
```

06/018

Connected to remote server/host zos18 using lu/pool SC0TCP14 and port 23

To verify installation and start the CONNX Listener process

After the CONNX installation runs to completion, the first verification step is to check that all job steps completed with condition codes less than or equal to four. A successful installation of the CONNX for Adabas components creates data sets with the default user-modifiable prefix of CONNX.ADABAS. The minimal complement of data sets necessary to run CONNX for Adabas, given the default data set prefix of CONNX.ADABAS, is as follows:

CONNX.ADABAS.CNTL
CONNX.ADABAS.OBJECT
CONNX.ADABAS.LOAD

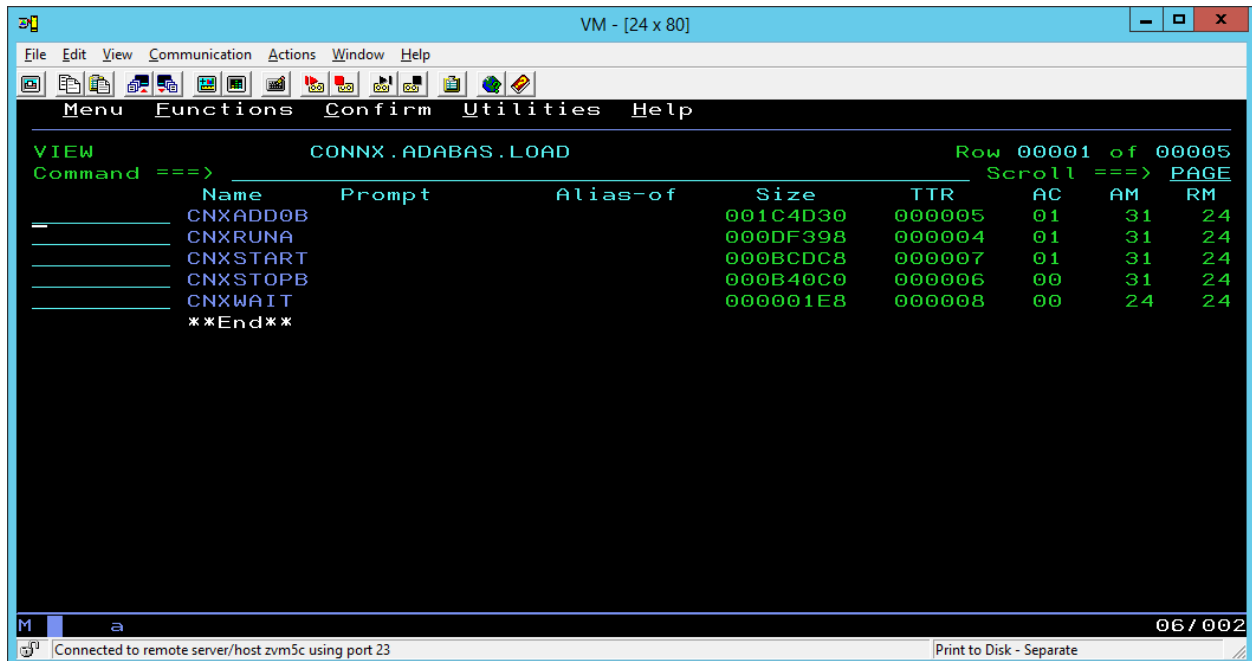
The following sections enumerate the manual steps necessary to complete the installation of the started task components of CONNX for Adabas.

To verify installation of CONNX Started Task components

- Given the default installation prefix of CONNX.ADABAS, the following data sets are created by a successful complete installation job:

CONNX.ADABAS.CNTL
CONNX.ADABAS.LOAD
CONNX.ADABAS.OBJECT

- Browse the CONNX.ADABAS.LOAD PDS to ensure that the following load modules are present and executable:



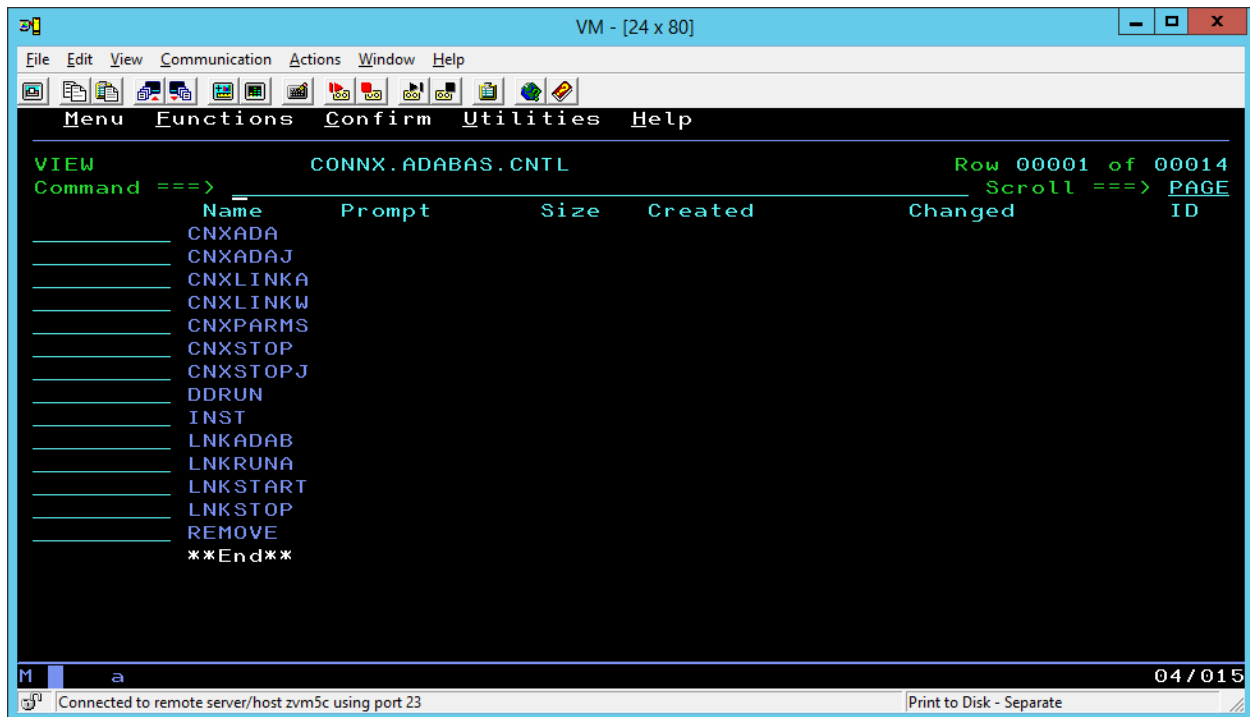
Member	Purpose
CNXAD'X'0B	The CONNX for Adabas TCP/IP Started Task /

	Batch Job Server Program. An instance of this program is created for each PC-to-OS/390 / z/OS server TCP/IP connection. 'X' is the version number (10=A, 11=B, 12=C, 13=D, etc)
CNXRUNA	The CONNX for Adabas Started Task / Batch Job TCP/IP Listener program. This program listens for PC client connection requests and starts an instance of server program CNXADD0B for each request.
CNXSTART	The CONNX for Adabas top-level 'broker' program. This is the main program in the CONNX for whatever Started Task / Batch Job address space. The CNXSTART program starts an instance of the CNXRUNA Started Task / Batch Job TCP/IP Listener program and an instance of the OS/390 / z/OS operator console command program CNXWAIT.
CNXSTOPB	The CONNX for Adabas TSO / Batch Job TCP/IP Listener stop program. This program can be executed via the TSO CNXSTOP REXX comamnd file in the CONNX.**.CNTL partitioned data set or via batch JCL to gracefully terminate the CONNX TCP/IP Listener program (CNXRUNA).
CNXWAIT	The CONNX for Adabas OS/390 / z/OS operator console interface program. Launched by program CNXSTART during the startup phase of the main job step of the CONNX for whatever Started Task / Batch Job address space. This program intercepts OS/390 / z/OS operator console /STOP jobname commands and gracefully terminates the CONNX for whatever Started Task / Batch Job address space, including the CONNX TCP/IP Listener program (CNXRUNA) and its associated instance(s) / subtask(s) of the CONNX for whatever Started Task / Batch Job Server Program (CNXADD0B).

Note: CNXRUNA and CNXAD'X'0B are built with AC flag =1, which means that they must reside in an APF-authorized load library. This requirement can be temporarily deferred by changing a CONNX environment variable. By default, the CONNX Started Task TCP/IP Listener program (CNXRUNA) authenticates user ID/password entries received from the CONNX client layer by calling the IBM Resource Access Control Facility (RACF) VERIFY exit. To bypass this logic and the APF-authorized load library requirement, define the following CONNX environment variable / value pair:

CNXNOPREAUTHORIZE=1

3. Refer to member CNXPARDS of CONNX.ADABAS.CNTL for the default CONNX parameter settings.



Member	Purpose
INST	The CONNX for Adabas installation job built by the CONNX Windows GUI installation program.
REMOVE	The CONNX for Adabas installation job + delete steps for the CONNX for Adabas sample files.
CMPCUST	ADAMP control records to compress the CONNX sample Customer file.
CMPEQUIP	Control records to compress the CONNX sample Equipment file.
CMPPORD	Control records to compress the CONNX sample Orders file.
CMPPGRP	Control records to compress the CONNX sample Product Group file.
CMPPROD	Control records to compress the CONNX sample Product file.
CNXADA	JCL procedure to invoke the CONNX for Adabas TCP/IP Listener/Server programs.
CNXADAJ	Execution JCL for procedure CNXADA.
CNXLINKA	JCL procedure to execute the IBM linkage editor against CONNX for Adabas object code and create the CONNX load modules.
CNXPARGS	Sample control records which define the environment variables for the CONNX for Adabas TCP/IP Listener and Server programs (CNXRUNA and CNXADD0B).
CNXSTOP	REXX command file to invoke program CNXSTOPB to stop CONNX TCP/IP Listener Program(s).
LNKADAB	Linkage editor control records for load module CNXADD0B.

LNKRUNA	Linkage editor control records for load module CNXRUNA.
LNKSTOP	Linkage editor control records for load module CNXSTOPB.

4. Browse CONNX.ADABAS.CNTL and find members CNXADA and CNXADAJ. Member CNXADA defines a JCL procedure which invokes the CONNX for Adabas TCP/IP Listener/Server programs. The batch job execution JCL for this procedure is in member CNXADAJ. Review the following procedure parameters:

Parameter	Definition
ADALOAD	Fully qualified Adabas Nucleus load library dataset name
CNXHLQ	High-level qualifier for CONNX JCL and load library partitioned data sets
CNXPARMS	Member name for CONNX environment variable control records
TCPHLQ	High-level qualifier for TCP/IP load libraries
REG	Job step region parameter
SYSLMOD	Fully qualified CONNX for Adabas load library dataset name

The parameters are preset with values entered from the CONNX for Adabas Windows GUI Installation program. Member CNXADAJ contains commented-out overrides for these parameters.

For testing purposes, edit member CNXPARMS of CONNX.ADABAS.CNTL and review the pre-set CONNX configuration settings. For more information, see the CONNX Configuration Settings chapter in the *CONNX User Reference Guide*.

6. To start the CONNX Listener, edit member CNXADAJ, change the job card parameters to match your installation standards, and submit the job to a batch execution queue. The CONNX TCP/IP Listener program (CNXRUNA) starts execution and attempts to listen on the TCP/IP port defined by the CNXRUNPORT environment variable. The following CONNX debug tracing messages show a successful attempt to listen on port 6710:

```
Listening to network port 6710
Binding to port 6710
NET TCP Listening on (0.0.0.0) Port(6710)
```

To enable RACF security

Enabling the user ID/password verification logic requires that the CONNX.ADABAS.LOAD library be APF-authorized.

1. If you have systems programmer authority, you can enter the following operator command from the system console or from the SDSF command line:

```
SETPROG APF,ADD,DSNAME=CONNX.ADABAS.LOAD,VOLUME=volume
```

Where volume is the DASD volume serial on which CONNX.ADABAS.LOAD resides.

If this command succeeds, the following message is displayed in the system log:

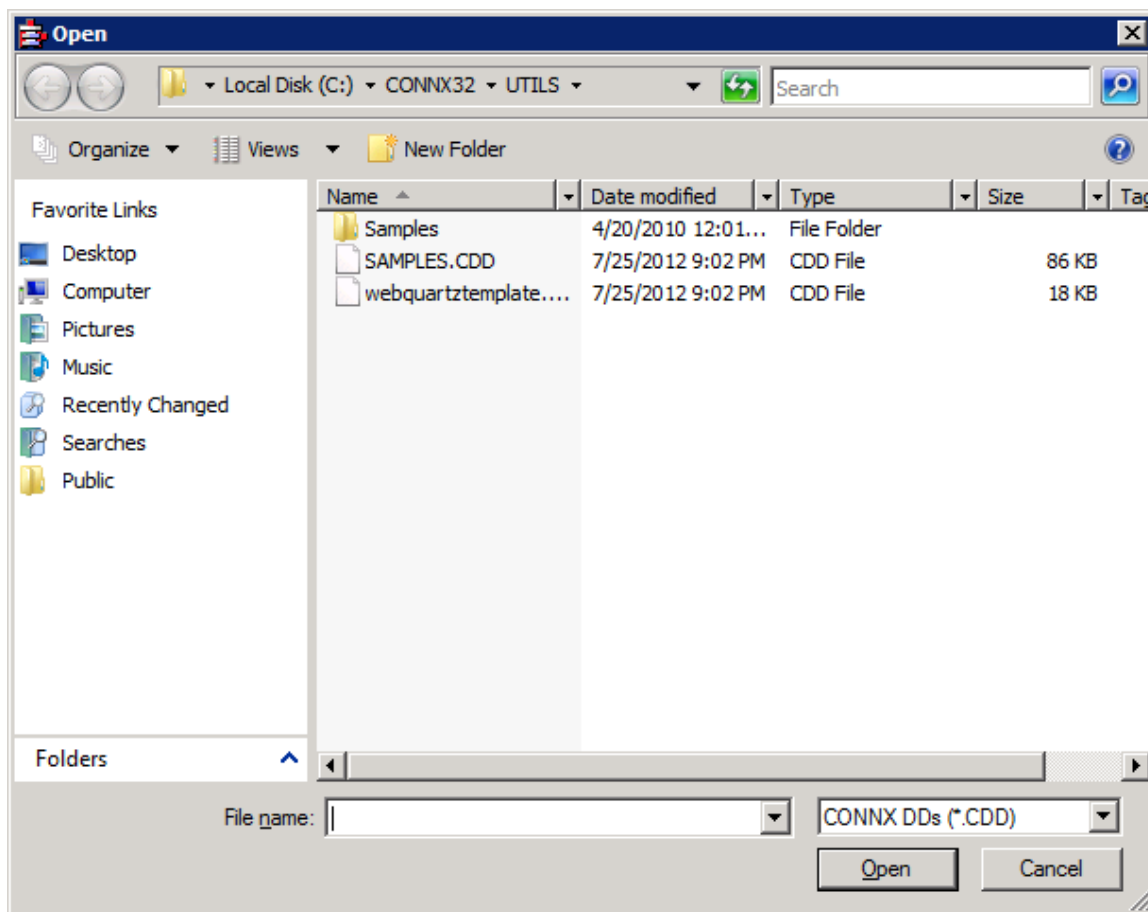
```
CSV410I DATA SET CONNX.ADABAS.LOAD ON VOLUME volume ADDED TO APF LIST
```

2. Edit member CNXPARGS of CONNX.ADABAS.CNTL and make the following CONNX environment variable entries:

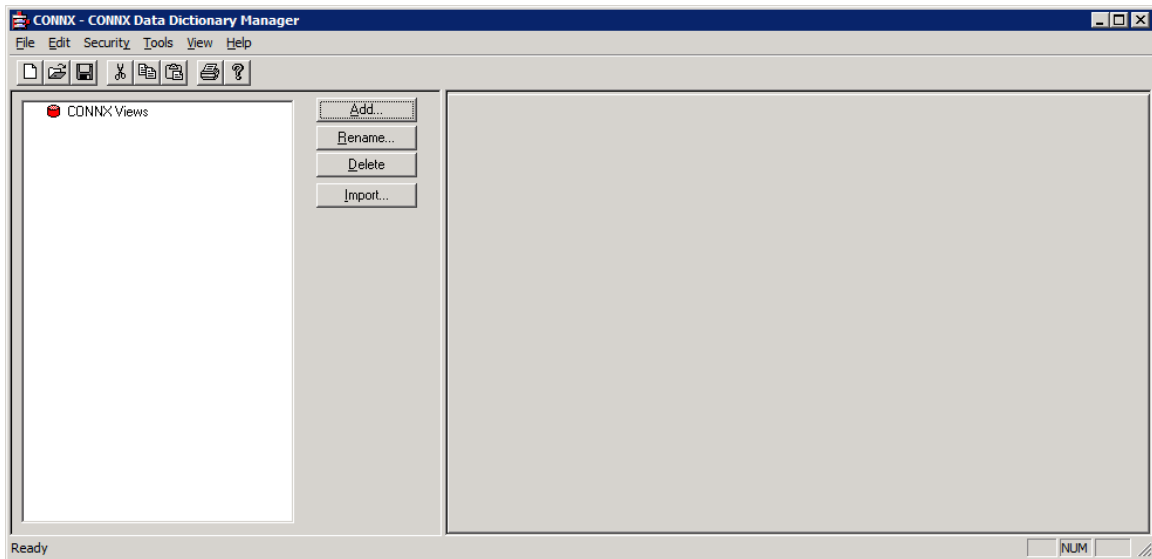
```
CNXNOPREAUTHORIZE=0
```

To test the CONNX for Adabas user ID/password verification

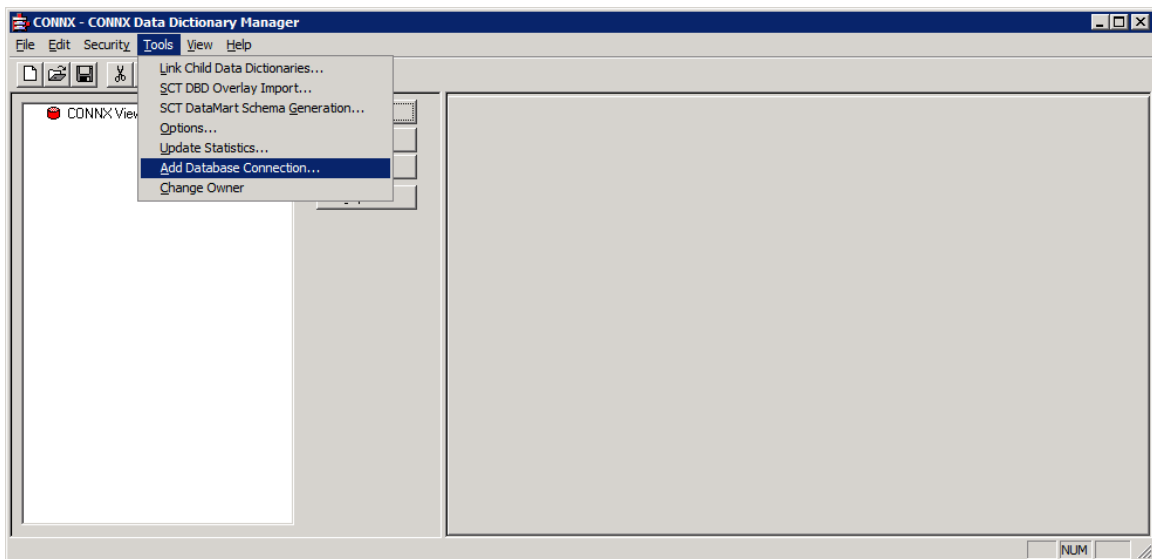
1. To test CONNX connectivity, edit member CNXADAJ, change the job card parameters to match your installation standards, and submit the job to a batch execution queue. Verify that the CONNX Listener Program CNXRUNA successfully binds to the specified =CNXRUNPORT value.
2. From the Windows **Start** menu, point to **All Programs**, point to **CONNX Solutions**, point to **CONNX**, and then click **CONNX Data Dictionary**.
3. The CONNX Data Dictionary Manager window and the Open dialog box appear. Select Cancel from this dialog box.



4. The CONNX Data Dictionary can be used to create a test connection..



5. From the Tools menu, select Add Database Connection....



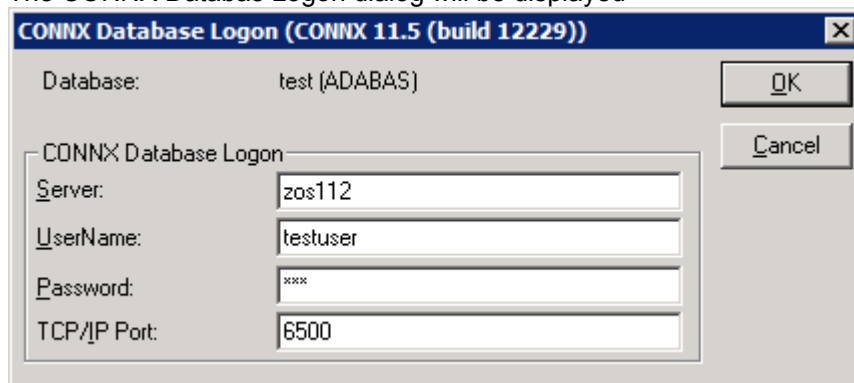
6. Enter a Database Name for this connection. For this example, use test. Select ADABAS in the Database Type drop down. The Server Name is the name (or IP address) of the mainframe where the Adabas listener was installed and the Databas ID is the DBID of a started Adabas nucleus.

Enter the logical name of the new database:

Database Name:	<input type="text" value="test"/>	<input type="button" value="OK"/>
Database Type:	<input type="text" value="ADABAS"/>	<input type="button" value="Cancel"/>
Server Name:	<input type="text" value="zos112"/>	
Database ID:	<input type="text" value="1"/>	

Press OK when all the fields have been filled in.

7. The CONNX Databas Logon dialog will be displayed



The image shows a dialog box titled "CONNX Database Logon (CONNX 11.5 (build 12229))". It contains the following fields and controls:

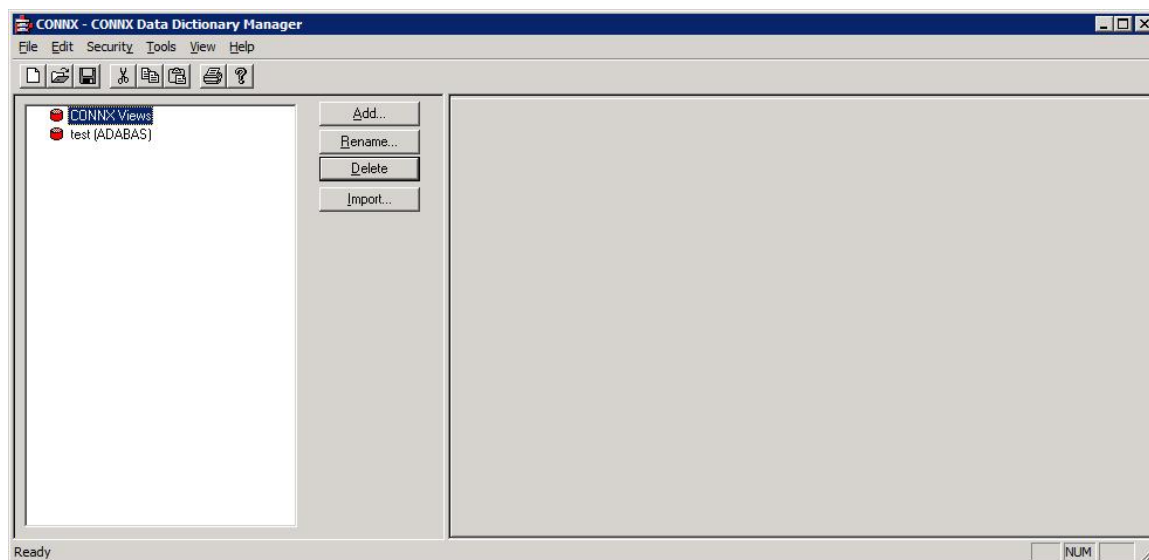
- Database:** A text field containing "test (ADABAS)".
- CONNX Database Logon:** A group box containing:
 - Server:** A text field containing "zos112".
 - UserName:** A text field containing "testuser".
 - Password:** A text field containing "xxx".
 - TCP/IP Port:** A text field containing "6500".
- Buttons:** "OK" and "Cancel" buttons are located on the right side of the dialog.

The Server name will be filled in from the previous dialog.

Enter a valid RACF User Name and password.

In the TCP/IP port text box, enter the numeric TCP/IP port on which the CONNX for Adabas TCP/IP listener program is currently listening. This entry must match the **=CNXRUNPORT=nnnn** entry in the CONNX.ADABAS.CNTL(CNXPARDS), where nnnn is a numeric TCP/IP port. The default value for the TCP/IP port is 6500.

Press the OK button. If everything is correct, you will return to the CONNX Data Dictionary main screen and see an entry named Test.



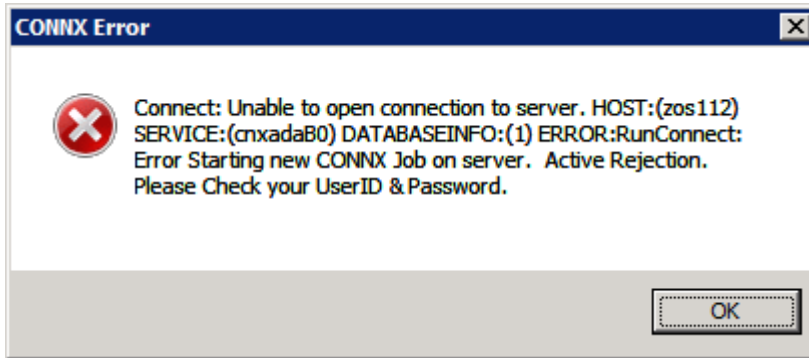
The image shows the main window of the "CONNX - CONNX Data Dictionary Manager". It features a menu bar (File, Edit, Security, Tools, View, Help) and a toolbar with icons for file operations. The main area is divided into two panes:

- Left Pane:** A list box containing two entries: "CONNX Views" and "test (ADABAS)".
- Right Pane:** A large empty area for displaying details.
- Buttons:** "Add...", "Rename...", "Delete", and "Import..." buttons are located between the two panes.
- Status Bar:** At the bottom, it shows "Ready" on the left and "NUM" on the right.

Pressing OK from the CONNX Database Logon screen makes a connection to the listener on the mainframe and verifies the user id and password. If the user id or password is incorrect or if the listener is not running on the specified port, the connection will fail.

8. After successfully connecting via the Add Database Connection option, repeat Steps5-7, but specify a non-existent user ID/password combination. In this case, an Active Rejection error will

be displayed. **Note:** Prior to repeating Steps 5-7, select the database named test and then press the Delete button.



If you do not see the above Active Rejection error message when using an incorrect User ID/Password combination, check to make sure that the parameter CNXNOPREAUTHORIZE is not set to 1 in CONNX.ADABAS.CNTL(CNXPARMS) on the mainframe.

9. Exit the CONNX Data Dictionary application. There is no need to save the changes to the CDD that was created for this process..
10. **Optional installation step:**
After verifying the JCL procedure CNXADA executes correctly as a batch job, you can copy it to a system started procedure partitioned data set, for example, SYS1.PROCLIB. As a final installation step, ask your security administrator or systems programmer to create a RACF user ID for the CNXADA procedure.

Note: *There are many advantages to creating CONNX started procedures, but this step can be bypassed if the alternative of running CONNX as one or more batch jobs provides adequate response time and security.*

To stop the CONNX for Adabas Batch Job / Started Task TCP/IP Listener

To stop the CONNX for Adabas TCP/IP Listener gracefully, use one of the following procedures:

1. Stop the CONNX for Adabas TCP/IP Listener with the /STOP operator console command.
2. Stop the CONNX for Adabas TCP/IP Listener via the CNXSTOP REXX command.
3. Stop the CONNX for Adabas TCP/IP Listener by submitting batch job CNXSTOP from member CNXSTOPJ of the CONNX.**.CNTL partitioned data set.

Important: Stopping the CONNX for Adabas TCP/IP Listener program CNXRUNA causes its started task or batch job to exit. When CNXRUNA terminates, it halts all instances of the CONNX for Adabas server program (CNXADD0B), which run as sub-tasks in the same address space. A separate instance (subtask) of CNXADD0B is created for each CONNX client connection via ODBC, JDBC, OLE DB or .NET to the specified TCP/IP address + port. Therefore, stopping the CNXRUNA program severs any active CONNX connections to the CNXADD0B server subtasks.

1. To stop the CONNX for Adabas TCP/IP Listener with the /STOP operator console command

Prerequisites:

1. You must be authorized to issue operator console commands.
2. The CONNX top level 'broker' program (CNXSTART) must be the main program of job step CNXRUNA in procedure CNXADA. This is the default setting, which can be overridden at JCL-submission time by changing the symbolic parameter CNXSTART.
3. The environment variables CNXLISTENER and CNXOPERATORCOMMAND must be defined as follows in member CNXPAMS of the CONNX.**.CNTL partitioned data set:

```
=CNXLISTENER=CNXRUNA
=CNXOPERATORCOMMAND=1
```

Assume that batch job CNXADA is executing JCL procedure CNXADA from the CONNX.**.CNTL partitioned data set, and that four PC clients are currently connected to separate instances of the CONNX for Adabas TCP/IP server program. To stop the CONNX for Adabas TCP/IP Listener via an operator console command, enter `/stop cnxada` as shown below:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 0 SIO 2 CPU 25 LINE 1-1 (1)
COMMAND INPUT ==> /stop cnxada SCROLL ==> CSR
PREFIX=CNXADA DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAME PROCSTEP JOBID OWNER C POS DP PGN REAL PAGING S
CNXADA CNXADA CNXRUNA JOB05544 IBMUSER A LO FF 1 1291 0.00 0.

```

The CNXADA job terminates the CONNX TCP/IP Listener program (CNXRUNA), as well as all of its currently executing server program (CNXADD0B) subtask(s). After the job completes, for each currently executing server program subtask, message pairs such as the following are written to the job log:

```
BPXP009I THREAD 0878136000000002, IN PROCESS 16842832, ENDED
ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000.
```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY CNXADA  JOB05544  DSID      2 LINE 0          COLUMNS 02- 81
COMMAND INPUT ==> _          SCROLL ==> CSR
*****
JES2 JOB LOG -- SYSTEM SYS1 -- N
-----
14.17.04 JOB05544  ---- TUESDAY, 04 MAY 2004 ----
14.17.04 JOB05544  IRR010I  USERID IBMUSER IS ASSIGNED TO THIS JOB.
14.17.07 JOB05544  ICH7000I  IBMUSER LAST ACCESS AT 14:06:54 ON TUESDAY, MAY 4,
14.17.07 JOB05544  $HASP373 CNXADA  STARTED - INIT 2  - CLASS A - SYS SYS1
14.17.07 JOB05544  IEF403I  CNXADA - STARTED - TIME=14.17.07
14.20.47 JOB05544  BPXP009I  THREAD 0878136000000002, IN PROCESS 16842832, ENDED
14.20.47 JOB05544  ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000
14.20.48 JOB05544  BPXP009I  THREAD 08785AA000000003, IN PROCESS 16842832, ENDED
14.20.48 JOB05544  ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000
14.20.49 JOB05544  BPXP009I  THREAD 0878668000000004, IN PROCESS 16842832, ENDED
14.20.49 JOB05544  ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000
14.20.50 JOB05544  BPXP009I  THREAD 0878726000000005, IN PROCESS 16842832, ENDED
14.20.50 JOB05544  ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000
14.20.51 JOB05544  IEF404I  CNXADA - ENDED - TIME=14.20.51
14.20.52 JOB05544  $HASP395 CNXADA  ENDED
----- JES2 JOB STATISTICS -----
MA a 04/021

```

2. To stop the CONNX for Adabas TCP/IP Listener via the CNXSTOP REXX command

Prerequisites:

1. The CONNX top level 'broker' program (CNXSTART) must be the main program of job step CNXRUNA in procedure CNXADA. This is the default setting, which can be overridden at JCL-submission time by changing the symbolic parameter CNXSTART.
2. The environment variables CNXLISTENER and CNXOPERATORCOMMAND must be defined as follows in member CNXPARMs of the CONNX.**.CNTL partitioned data set:

```
=CNXLISTENER=CNXRUNA
=CNXOPERATORCOMMAND=0
```

3. For example, assume that the CONNX for Adabas TCP/IP listener (CNXADAJ, JOB01099) is started and listening on port 6789.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 0 SID 12 CPU 100 LINE 1-1 (1)
COMMAND INPUT ==> SCROLL ==> CSR
PREFIX=CNX* DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAME PROCSTEP JOBID OWNER C POS DP PGN REAL PAGING S
CNXADAJ CNXADA CNXRUNA JOB01099 IBMUSER A LD FF 1 446 0.00 0.
-----
MA a 04/021

```

To stop the TCP/IP Listener program CNXRUNA and end the job, use the CNXSTOP REXX command. The verbose and implicit syntax and an example of the implicit execution of this command follow:

From the ISPF Command Shell (default option 6):

```

cnxstop 6789
or
exec 'connx.adabas.cntl(cnxstop)' 6789

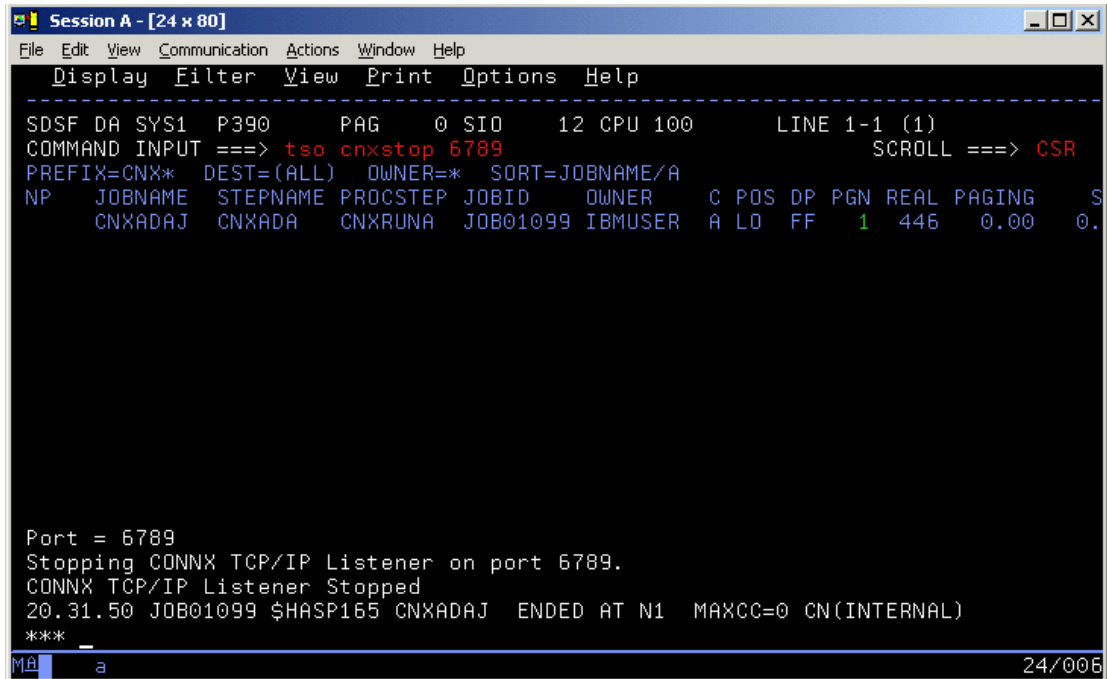
```

From any ISPF Command line:

```

tso cnxstop 6789
or
tso exec 'connx.adabas.cntl(cnxstop)' 6789

```



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 0 SIO 12 CPU 100 LINE 1-1 (1)
COMMAND INPUT ==> tso cnxstop 6789 SCROLL ==> CSR
PREFIX=CNX* DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAME PROCSTEP JOBID OWNER C POS DP PGN REAL PAGING S
CNXADAJ CNXADA CNXRUNA JOB01099 IBMUSER A LO FF 1 446 0.00 0.

Port = 6789
Stopping CONNX TCP/IP Listener on port 6789.
CONNX TCP/IP Listener Stopped
20.31.50 JOB01099 $HASP165 CNXADAJ ENDED AT N1 MAXCC=0 CN(INTERNAL)
***
MA a 24/006
  
```

3. To stop the CONNX for Adabas TCP/IP Listener by submitting batch job CNXSTOP from member CNXSTOPJ of the CONNX.**.CNTL partitioned data set

Prerequisites:

1. The CONNX top level 'broker' program (CNXSTART) must be the main program of job step CNXRUNA in procedure CNXADA. This is the default setting, which can be overridden at JCL-submission time by changing the symbolic parameter CNXSTART.
2. The environment variables CNXLISTENER and CNXOPERATORCOMMAND must be defined as follows in member CNXPARMS of the CONNX.**.CNTL partitioned data set:

```

=CNXLISTENER=CNXRUNA
=CNXOPERATORCOMMAND=0
  
```

For example, assume that the CONNX for Adabas TCP/IP listener (CNXADAJ, JOB05559) is started and listening on port 6789:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

File Edit Confirm Menu Utilities Compilers Test Help

EDIT      CONNX.ADAD9008.CNTL(CNXSTOPJ) - 01.00      Columns 00001 00072
Command ==> sub      Scroll ==> CSR
***** ***** Top of Data *****
000001 //CNXSTOPJ JOB (ACCT#),
000002 //      'CONNX 9.0',
000003 //      CLASS=A,
000004 //      MSGCLASS=H,
000005 //      MSGLEVEL=(1,1),
000006 //      LINES=(50),
000007 //      NOTIFY=&SYSUID ,
000008 //*      USER=,
000009 //*      PASSWORD=,
000010 //*      REGION=2M
000011 //*      TIME=(2,59),
000012 //*
000013 //*----- CONNX Version 9.0 -----*
000014 //*----- Copyright 2004 by CONNX Solutions Inc. -----*
IKJ56250I JOB CNXSTOPJ(JOB05560) SUBMITTED
***  _
MA      a      22/006

```

To stop the TCP/IP Listener program and end the job, edit member CNXSTOPJ in the CONNX.**.CNTL partitioned data set, change the job card to match site-specific standards, and submit the job to a batch execution queue.

Both jobs CNXADAJ and CNXSTOPJ should end with condition codes = 0:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

19.08.53 JOB05560 $HASP165 CNXSTOPJ ENDED AT N1 MAXCC=0 CN(INTERNAL)
19.08.55 JOB05559 $HASP165 CNXADAJ ENDED AT N1 MAXCC=0 CN(INTERNAL)
***  _
MA      a      03/006

```

The CNXADAJ job terminates the CONNX TCP/IP Listener program (CNXRUNA), as well as all of its currently executing server program (CNXAD0B) subtask(s). After the job completes, for each currently-executing server program subtask, message pairs such as the following are written to the job log:

```

BPXP009I THREAD 08785AA000000002, IN PROCESS 83951697, ENDED
ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000.

```

Step 5 - Install IBM Mainframe-Compatible Server

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY CNXADAJ JOB05559 DSID      2 LINE 0      COLUMNS 02- 81
COMMAND INPUT ==> _      SCROLL ==> CSR
***** TOP OF DATA *****
              J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  --  N

18.59.59 JOB05559 ---- TUESDAY, 04 MAY 2004 ----
18.59.59 JOB05559 IRR010I USERID IBMUSER IS ASSIGNED TO THIS JOB.
19.00.00 JOB05559 ICH70001I IBMUSER LAST ACCESS AT 18:54:33 ON TUESDAY, MAY 4,
19.00.01 JOB05559 $HASP373 CNXADAJ STARTED - INIT 2 - CLASS A - SYS SYS1
19.00.01 JOB05559 IEF403I CNXADAJ - STARTED - TIME=19.00.01
19.08.52 JOB05559 BPXP009I THREAD 08785AA000000002, IN PROCESS 83951697, ENDED
ABNORMALLY WITH COMPLETION CODE 0013E000, REASON CODE 00000000
19.08.54 JOB05559 IEF404I CNXADAJ - ENDED - TIME=19.08.54
19.08.54 JOB05559 $HASP395 CNXADAJ ENDED
----- JES2 JOB STATISTICS -----
04 MAY 2004 JOB EXECUTION DATE
      27 CARDS READ
      174 SYSOUT PRINT RECORDS
        0 SYSOUT PUNCH RECORDS
        10 SYSOUT SPOOL KBYTES
      8.89 MINUTES EXECUTION TIME
. . . . .
MA a 04/021

```

SAS/C Network Administration

The operation of the SAS/C socket library depends on its ability to access the configuration information for a site. In some cases, the socket library located the information automatically. In other cases, it may be necessary to specify the location of the information.

The location and usage of site configuration files are described in the SAS/C Library Reference, Volume 2, Release 7.00, Part 2, SAS/C Socket Library for TCP/IP, section "Network Administration".

To determine the TCP/IP stack

In order to determine the TCP/IP stack that is to be used by the Adabas SQL Gateway data server, you must add an assignment for the SAS/C environment variables 'TCPIP_MACH' to the CNXPARMS member described in [To verify installation and start the CONNX Listener process.](#)

For IBM TCP/IP, the socket library must locate the TCP/IP address spaced under z/OS. The name of the address space may vary from site to site.

The SAS/C compiler uses the TCPIP_MACH environment variable to determine the value of this name. If the TCPIP_MACH variable does not exist, the socket library searches for the name in the TCPIP.DATA files under z/OS. If this file is not available, the socket library uses a default value of TCPIP to locate the TCP/IP address space.

For more information, refer to

<http://support.sas.com/documentation/onlinedoc/sasc/doc700/html/lr2/zr2netma.htm>

Part 3 - Uninstalling the Adabas SQL Gateway (CONNX for Adabas) components

Uninstalling the Adabas SQL Gateway (CONNX for Adabas) components

The CONNX for Adabas TCP/IP Server Setup client application transfers the CONNX components to the target OS/390 / z/OS host via TCP FTP. After the initial FTP step, the CONNX installation continues via a batch job submission using TSO ISPF. Final optional installation steps for the CONNX for Adabas Started Task components include manual additions of a JCL procedure (CNXADA) and a REXX exec file (CNXSTOP) to the system PROCLIB and SYSPROC partitioned data sets.

Given a default CONNX installation prefix of CONNX.ADABAS, the CONNX for Adabas Started Task installation JCL partitioned data set is CONNX.ADABAS.CNTL. Member REMOVE contains batch uninstallation JCL.

To uninstall the CONNX for Adabas Started Task components

1. Before submitting the uninstallation JCL, find and stop all instances of the CONNX for Adabas Started Task/Batch Job TCP/IP Listener program CNXRUNA. For example, batch job CNXADAJ is executing, listening on TCP/IP port 6789.
2. Use the TSO NETSTAT command to list the TCP/IP listen port dedicated to CNXADAJ, and all active client/server connections:

```
EZZ2587I CNXADAJ 0009C426 123.123.123.123..1927 123.123.123.124..4397
      Establish
```

```
EZZ2587I CNXADAJ 0009C424 0.0.0.0..6789          0.0.0.0..0
      Listen
```

```
EZZ2587I CNXADAJ 0009C42D 123.123.123.123..1930 123.123.123.125..4405
      Establish
```

```
EZZ2587I CNXADAJ 0009C429 123.123.123.123..1928 123.123.123.126..4401
      Establish
```

```
EZZ2587I CNXADAJ 0009C42B 123.123.123.123..1929 123.123.123.127..4403
      Establish
```

3. Use the TSO CNXSTOP command, specifying port 6789, to terminate the CNXRUNA listener program and its active subtasks.
4. Repeat these steps for other any other batch jobs/started tasks which are currently executing the CONNX TCP/IP listener program CNXRUNA.
5. After stopping all instances of the CNXADA Started Task(s)/Batch Job(s), you can submit the JCL in member REMOVE. Note that the last control record of the IDCAMS DELETE step references the CONNX.ADABAS.CNTL partitioned data set, so be sure to submit the job and deallocate the partitioned data set from your TSO session by navigating to another ISPF option.

Also note that the second IDCAMS DELETE step for the CONNX.ADABAS.LOAD partitioned data set is commented out. This is by design, in case the CONNX Started Task/Batch job load modules were linked into an existing load library. It is recommended, but not required, to install CONNX into a separate load library. If the CONNX load modules were installed into a new load library:

```

VM - [24 x 80]
File Edit View Communication Actions Window Help
Menu Functions Confirm Utilities Help

VIEW          CONNX.ADABAS.LOAD          Row 00001 of 00005
Command ==>
Name          Prompt          Alias-of          Size          TTR          AC          AM          RM
-----
CNXADD0B      001C4D30      000005          01          31          24
CNXRUNA       000DF398      000004          01          31          24
CNXSTART      000BCDC8      000007          01          31          24
CNXSTOPB      000B40C0      000006          00          31          24
CNXWAIT       000001E8      000008          00          24          24
**End**

M a 06/002
Connected to remote server/host zvm5c using port 23
Print to Disk - Separate

```

Then Step 2 can be commented in:

```

VM - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          CONNX.ADABAS.CNTL (REMOVE) - 01.00 Columns 00001 00072
Command ==>
000053 //STEP1      EXEC PGM=IDCAMS
000054 //SYSPRINT DD SYSOUT=*
000055 //SYSIN      DD *
000056 DELETE CONNX.ADABAS.CNTL      PURGE SCRATCH
000057 DELETE CONNX.ADABAS.OBJECT    PURGE SCRATCH
000058 /*
000059 /***STEP2      EXEC PGM=IDCAMS
000060 /***SYSPRINT DD SYSOUT=*
000061 /***SYSIN      DD *
000062 /*** DELETE CONNX.ADABAS.LOAD    PURGE SCRATCH
000063 //
***** ***** Bottom of Data *****

M a 04/015
Connected to remote server/host zvm5c using port 23
Print to Disk - Separate

```

6. The IDCAMS DELETE steps should return zeroes for all data sets which are not allocated to another batch job, started task, or TSO session. For data sets which do not exist or are allocated to other jobs or tasks, the IDCAMS return code is 8. The undeleted data sets, if any, can be deleted via ISPF or via a second run of the REMOVE JCL after the dataset sharing conflicts have been resolved.

Manual Steps: Uninstallation

To verify that all CONNX for Adabas components have been uninstalled, use ISPF option 3.4 to list and delete the datasets defined with the (default) CONNX.ADABAS high-level qualifier. Next, delete all versions of the CNXADA procedures from the production JCL PROCLIB. Finally, if the CNXSTOP REXX exec file has been copied to a SYSPROC PDS such as SYS1.LOCAL.CLIST, delete the CNXSTOP member.

Adabas SQL Gateway (CONNX for Adabas) Security

Host / Client Security Overview

CONNX for Adabas is implemented as a Windows PC client and a pair of OS/390 / z/OS programs: the Listener (CNXRUNA) and the Server (CNXRUNB). A CONNX for Adabas user connects from a client application through a supported interface (ODBC, JDBC, OLE DB, or .NET) to the CONNX TCP/IP Listener program CNXRUNA, which starts program CNXRUNB as a subtask in the CNXRUNA batch address space. Refer to steps 1 through 3 in the following Figure 1. Once started, program CNXRUNB sets up a separate TCP/IP connection to the invoking CONNX PC client.

Default Option

The host-side security logic for user ID / password verification can be enabled or disabled for CONNX client-server connections via a CONNX environment variable. Setting CNXNOPREAUTHORIZE to a non-zero value instructs the CONNX for Adabas TCP/IP Listener and Server programs to bypass user ID / password verification. Once the listener program starts a server program subtask, client requests and data flow from the CONNX PC client to the server subtask and back via a dedicated TCP/IP socket connection.

This security option bypasses the need to execute the CONNX programs from an APF-authorized load library. An advantage to this approach is that userid/password security rules for CONNX PC clients need only be defined for each OS/390 / z/OS user ID which starts program CNXRUNA.

Alternate Option

Setting CNXNOPREAUTHORIZE to zero instructs the CONNX for Adabas TCP/IP Listener and Server programs to execute the host-side user ID / password verification logic. The CONNX PC client component sends an encrypted user ID / password pair to the CONNX TCP/IP Listener program CNXRUNA, which decrypts the user ID / password and executes the RACF (Resource Access Control Facility) VERIFY macro. If the user ID / password pair is valid, program CNXRUNA starts program CNXRUNB as a subtask in the CNXRUNA batch address space. Refer to steps 1 through 3 in Figure 6-4. Once started, program CNXRUNB sets up a separate TCP/IP connection to the invoking CONNX PC client. Client requests and Adabas data responses flow back and forth directly from the CONNX PC client to the dedicated CNXRUNB subtask. Refer to steps 4 and 5 in the following Figure 2.

The alternate security option observes the host-side security rules defined for user ID / password verification. In order for these rules to be enforced, programs CNXRUNA and CNXRUNB must run from an APF (Authorized Program Facility) load library. The operator command (SETPROG APF) necessary to add the CONNX for Adabas load library to the APF list is documented in [Installation Verification](#).

CONNX Client-Side Security and Host-Side Security

For both options, the CONNX Administrator should consider taking advantage of the client-side security features implemented in the CONNX Data Dictionary (CDD). The first line of defense is only to import selected Adabas files into the CDD. Additionally, the CONNX Administrator can restrict file access based on CONNX user IDs and groups defined via the security menu features in the CONNX Data Dictionary Manager.

In some cases, a physical Adabas file is composed of multiple logical files or sub-files. The CONNX Data Dictionary Manager can be used to define and enforce security rules based on these logical files, as well as on the underlying physical data sets. Further, column- and row-level security can be implemented by defining one or more CONNX views against the imported Adabas files, and by authorizing individual or groups of CONNX user IDs to execute the CONNX client-side views.

The CONNX client-side approach to security complements and enhances host-side security products, such as IBM RACF, CA-ACF2, and CA-Top Secret. In all cases, the security rules defined in the CONNX Data Dictionary take precedence over the host-side security rules. For more information on CONNX

client-side security, refer to the CONNX User Reference Guide, available online, on the CONNX CD-ROM, and within the CONNX application.

Figure 1

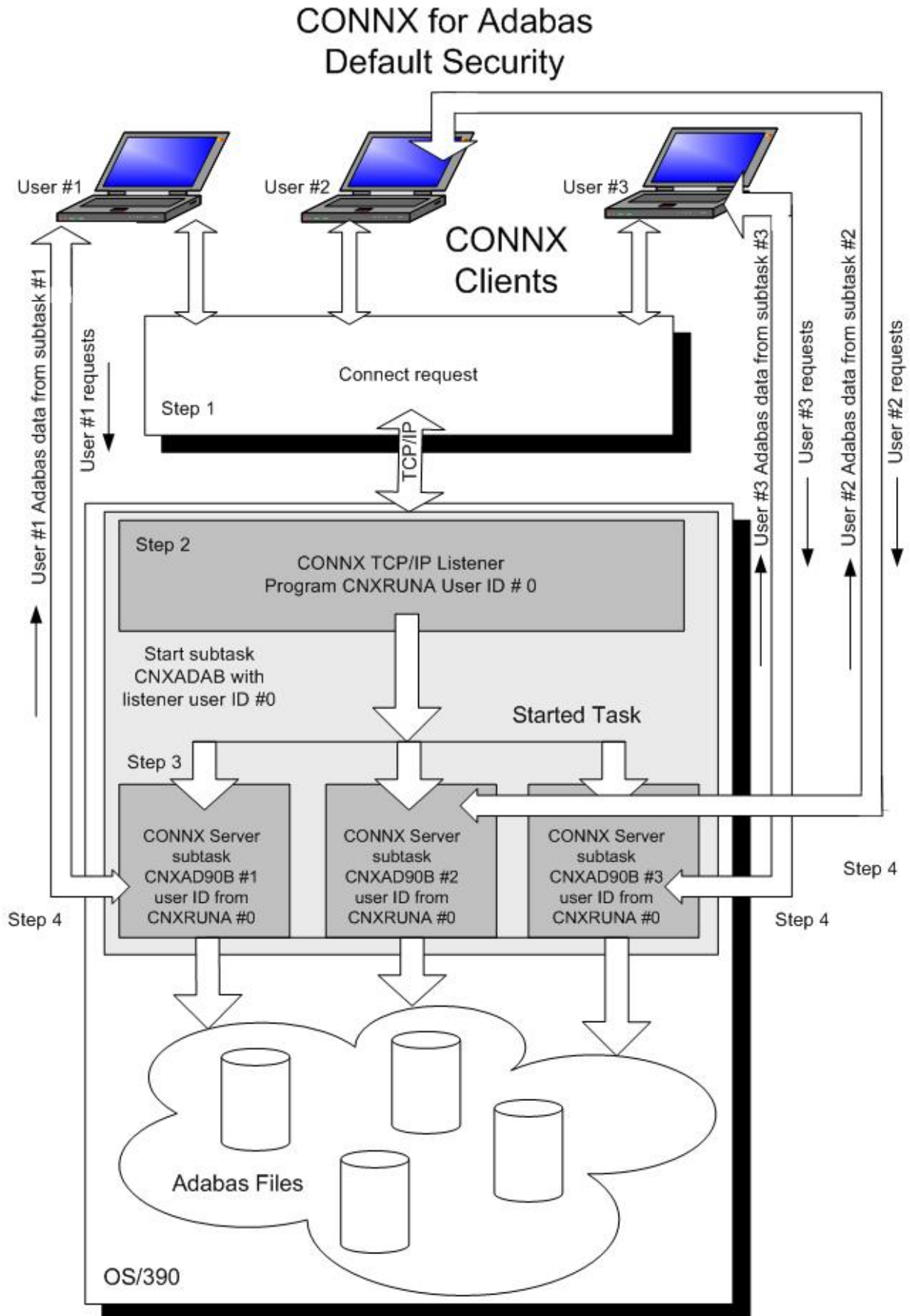
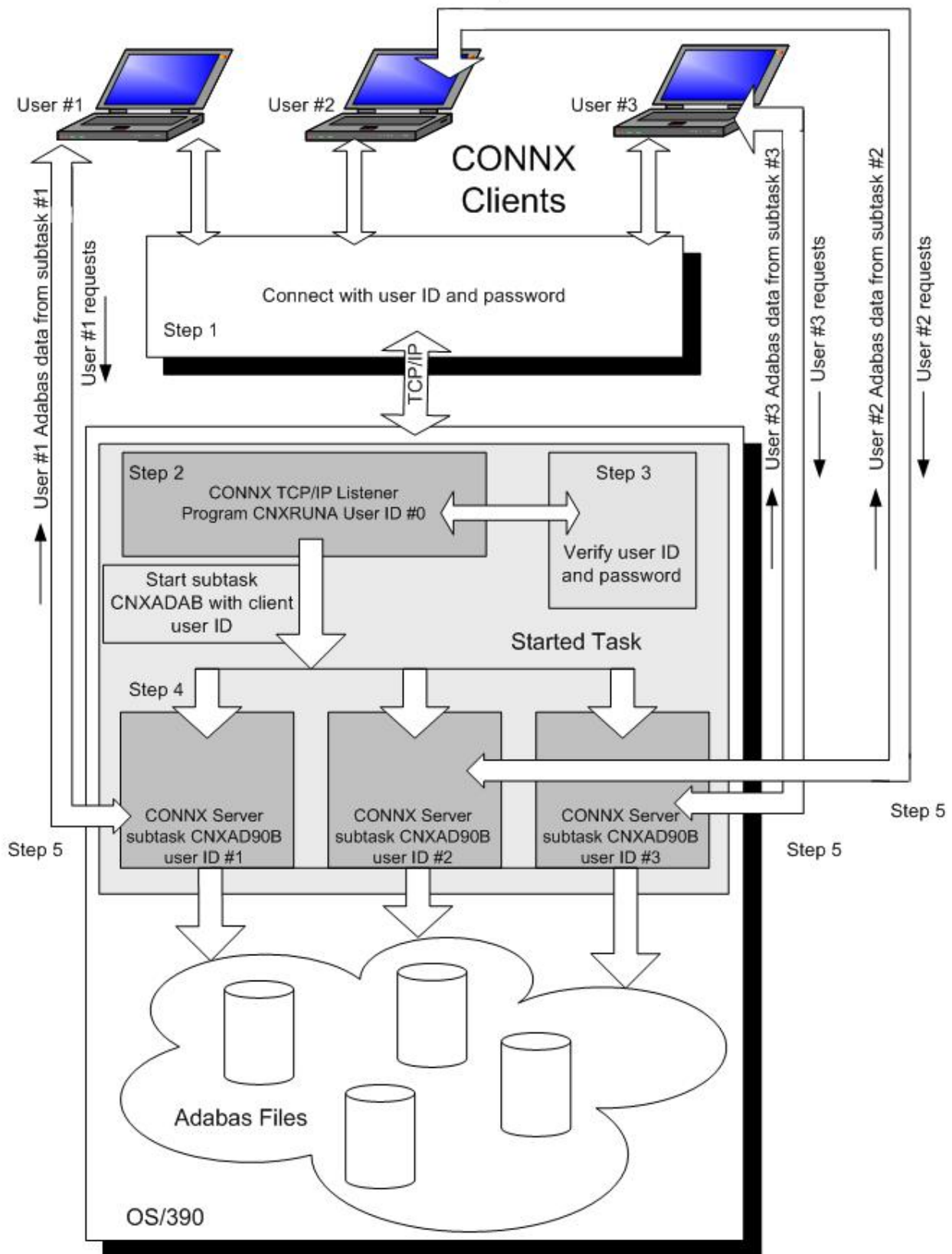


Figure 2

CONNX for Adabas Alternate Option



Adabas SAF Security

The usage of SAF Security can be activated “after the fact”; without re-installing the data server component. The following steps are required to enable SAF Security support on z/OS:

- APF-authorize the SQL Gateway load library.
- Update the CONNX Listener parameters.
- Edit the member CNXPARMS .
- Set CNXNOPREAUTHORIZE to "0".
- Set CNX_PASS_TICKETS to "1".
- Restart the z/OS data server.

Additionally, the following actions are also recommended:

- Define CDD password to restricted access to the CDD.
- Disable "CONNX Integrated Security" in CDD, when user accounts are not added to the CDD.
- Enable JESSPOOL-Access for all SQL Gateway users; the RACF default for JESSPOOL is (ACCESS=NONE).

IMS**Prerequisites for IMS - ODBA**

- Verify APF Authorization if required.
 - If your system has a security manager like RACF installed, and you wish for all sessions to be authorized, set CNXNOPREAUTHORIZE to 0.
 - If CNXNOPREAUTHORIZE is set to 0, the following is required:
 - Verify that IMS.RESLIBL and the optional IMSUTIL.SHPSLMD0 (zero) are APF authorized.
 - The CONNX load library will need to be APF authorized
- IMS must be configured for ODBA access. Please refer to IBM IMS manuals for details on configuring IMS for ODBA.
- CONNX uses the AERTDLI API. When connecting to a database, CONNX uses the APSB function which requires RRS be running.
- The TCP/IP started task must be running.
- The IBM IMS TCP/IP sockets interface, including IMS configuration files, programs, maps, and transactions, must be installed.
- Both the install PC client and the target host (OS/390 / z/OS) must have FTP enabled.
- CONNX fully supports the IMS add-on product *IMS Library Integrity Utilities for z/OS* to make the importing of metadata easier. If you wish to use this feature, verify that the utilities are installed on your system. Note: This feature is optional.
- Create a DRA table for each IMS subsystem that will be accessed by CONNX. Please take special note of the MAXTHRD parameter. This value should be set to the maximum number of concurrent CONNX connections you intend to make. Following is some sample JCL that can be used to create a DRA table:

```
//IBUSERC JOB ACCT,REGION=32M,MSGLEVEL=(1,1),
//          CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM LINES=9999
//*REGION=32M
//*
//*
//*****
**
/* IVP IMS 8.1
/*
/* SKELETON: DFSIXSE9
/*
/* FUNCTION: DEFINE DRA START-UP TABLE FOR USE BY CICS/DBCTL
//*****
**
//*
//*****@SCPYRT
**
//*
*
/* LICENSED MATERIALS - PROPERTY OF IBM
*
```

```

// *

*
// *   RESTRICTED MATERIALS OF IBM
*
// *

*
// *   5655-C56
*
// *

*
// *   (C) COPYRIGHT IBM CORP. 1989,2002
*
// *

*
// *****@ECPYRT
**
// *
// *****
**
//ASMDRA   PROC MBR=DFSIVP10
// *
//ASM      EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IMS810.ADFSMAC
//         DD DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD UNIT=3390,DISP=(,PASS,DELETE),SPACE=(CYL,(1,1)),
//         DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSUT1   DD UNIT=3390,DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=3390,DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1))
//SYSUT3   DD UNIT=(3390,SEP=(SYSLIB,SYSUT1,SYSUT2)),
//         DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1))
// *
//LKED     EXEC PGM=IEWL,COND=(0,LT,ASM),
//         PARM='NCAL,LET,LIST,XREF'
//SYSPRINT DD SYSOUT=*
//SYSLIN   DD DISP=(OLD,DELETE,DELETE),
//         DSN=*.ASM.SYSLIN,VOL=REF=*.ASM.SYSLIN
//SYSLMOD   DD DISP=SHR,
//         DSN=IMS810.SDFSRESL(&MBR)
//SYSUT1   DD UNIT=(3390,SEP=(SYSLMOD,SYSLIN)),
//         DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1))
//         PEND
// *****
**
// *
// * THE FOLLOWING STEP ASSEMBLES AND LINK-EDITS INTO IMS RESLIB

```

```

// * THE DEFAULT PARAMETERS USED BY THE IMS CONTROL REGION - IVP3
// *
// * DFSPZPIV EXEC PROC=ASMDRA, MBR=DFSPZPIV
// DFSPZP10 EXEC PROC=ASMDRA, MBR=DFSPZP10
// ASM.SYSIN DD *
PZP      TITLE 'DATABASE RESOURCE ADAPTER STARTUP PARAMETER TABLE'
* DFSPZP00 CSECT
DFSPZP10 CSECT
*****
*
*
*
*      MODULE NAME: DFSPZP00
*
*
*
*
*      DESCRIPTIVE NAME: DATABASE RESOURCE ADAPTER (DRA)
*
*      STARTUP PARAMETER TABLE.
*
*
*
*      FUNCTION: TO PROVIDE THE VARIOUS DEFINITIONAL PARAMETERS
*
*      FOR THE COORDINATOR CONTROL REGION. THIS
*
*      MODULE MAY BE ASSEMBLED BY A USER SPECIFYING
*
*      THEIR PARTICULAR NAMES, ETC. AND LINKED INTO
*
*      THE USER RESLIB AS DFSPZPXX . WHERE XX
*
*      IS EITHER 00 FOR THE DEFAULT, OR ANY OTHER ALPHA-
*
*      NUMERIC CHARACTERS.
*
*
*
*
*      KEYWORDS FOR THE DFSPRP MACRO:
*
*      DSECT=NO-A DSECT STATEMENT FOR PRP WILL NOT BE
*
*      GENERATED (LABEL DFSPRP WILL BE ON DS 0D).
*
*
*

```

```

*          FUNCLV=  DEFAULT (1). ADAPTER FUNCTIONAL LEVEL.
*
*
*
*          DDNAME=  1 TO 8 CHARACTER DD NAME TO BE USED WITH
*
*          DYNAMIC ALLOCATION OF THE DBCTL RESLIB.
*
*          DEFAULT (CCTLDD).
*
*
*
*          DSNAME=  1 TO 44 CHARACTER DATASET NAME OF THE
*
*          DBCTL RESLIB.
*
*          DEFAULT (IMS.SDFSRESL).
*
*
*
*          DBCTLID=XXXX-NAME OF THE DBCTL REGION
*
*          DEFAULT = SYS1
*
*
*
*          USERID=XXXXXXXX-NAME OF THE USER REGION
*
*
*
*          MINTHRD=XXX-MINIMUM NUMBER OF THREADS TO BE
*
*          AVAILABLE (MAXIMUM NUMBER IS 255)
*
*          DEFAULT = 1
*
*
*
*          MAXTHRD=XXX-MAXIMUM NUMBER OF THREADS TO BE
*
*          AVAILABLE (MAXIMUM NUMBER IS 255)
*
*          DEFAULT = 1
*
*
*

```

Step 5 - Install IBM Mainframe-Compatible Server

```

*          TIMER=XX-IDENTIFY TIMER VALUE IN SECONDS (DEFAULT 60)
*
*
*
*          FPBUF=XXX-NUMBER OF FAST PATH BUFFERS TO BE ALLOCATED
*
*          AND FIXED PER THREAD (DEFAULT 00)
*
*
*
*          FPBOF=XXX-NUMBER OF FAST PATH OVERFLOW BUFFERS TO BE
*
*          ALLOCATED PER THREAD (DEFAULT 00)
*
*
*
*          SOD=X-OUTPUT CLASS TO BE USED FOR SNAP DUMP OF
*
*          ABNORMAL THREAD TERMINATIONS (DEFAULT A)
*
*
*
*          TIMEOUT=XXX-DRATERM TIMEOUT VALUE IN SECONDS(DEFULT
60)*
*
*
*          CNBA=XXXX -   TOTAL FP NBA BUFFERS FOR CCTL
*
*
*
*
*          AGN=XXXXXXXX-1 TO 8 CHARACTER APPLICATION GROUP NAME
*
*
*
*          LOCATION: PRIVATE STORAGE, USER KEY
*
*
*
*          THIS MODULE CONTAINS NO EXECUTABLE CODE.
*

```

```

*
*
*****
*
      EJECT
      DFSPRP DSECT=NO,
                                X
      FUNCLV=1,          X      CCTL FUNCTION LEVEL
      X
      DDNAME=CCNXDD,      XXXXXXXX DDN FOR CCTL RESLIB
DYNALOC X
      DSNAME=IMS810.SDFSRESL,
                                X
      DBCTLID=IVP1,      NAME OF DBCTL REGION
      X
      USERID=,          XXXXXXXX NAME OF USER REGION
      X
      MINTHRD=001,      XXX      MINIMUM THREADS
      X
      MAXTHRD=100,      XXX      MAXIMUM THREADS
      X
      TIMER=60,          XX      IDENTIFY TIMER VALUE -
SECS X
      FPBUF=001,          XXX      FP FIXED BFRS PER THREAD
      X
      FPBOF=001,          XXX      FP OVFLW BFRS PER THREAD
      X
      SOD=H,          X      SNAP DUMP CLASS
      X
      TIMEOUT=060,      XXX      DRATERM TIMEOUT IN SECONDS
      X
      CNBA=001,          XXX      TOTAL FP NBA BFRS FOR CCTL
      X
      AGN=IVP          XXXXXXXX APPLICATION GROUP NAME
      END
// *
// *      DBCTLID=, XXXX      NAME OF DBCTL REGION
      X
// *      USERID=CNXODBA,      XXXXXXXX NAME OF USER REGION
      X

```

- The PCB is accessed by name and not by the relative PCB number within the PSB. The PCB can be named in two ways:
 1. Add a label to each PCB macro line (columns 1-8). We recommend you use this method.
 2. Add a PCBNAME= parameter to the PCB macro.
- Regenerate each of the modified PSBs so the PSBLIB reflects the identifier for each PCB.
- Set parameter SCHDTYP=PARALLEL (default is SERIAL) in macro APPLCTIN. This is an ODBA requirement. You can find APPLCTIN in IMSGEN PDS member DFSIXSC1. An example of DFSIXSC1 can be found at

http://www.ibm.com/support/knowledgecenter/SSEPH2_14.1.0/com.ibm.ims14.doc.ins/ims_dbcd_bctl.htm.

Refer to the following links for more information on connecting IMS to ODBA:

IMS Connectivity in an On Demand Environment: A Practical Guide to IMS Connectivity©

Document Number: SG24-6794-00

<http://www.redbooks.ibm.com/abstracts/sg246794.html>

Verifying TCP/IP Connectivity

Before you install CONNX on your IMS server, we recommend that you verify the TCP/IP connectivity to your OS/390 z/OS system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

`ping hostname`

where `hostname` is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Prerequisites for IMS - DLI

- Verify APF Authorization if required.
 - If your system has a security manager like RACF installed, and you wish for all sessions to be authorized, set CNXNOPREAUTHORIZE to 0.
 - If CNXNOPREAUTHORIZE is set to 0, the following is required:
 - Verify that IMS.RESLIBL and the optional IMSUTIL.SHPSLMD0 (zero) are APF authorized.
 - The CONNX load library will need to be APF authorized
- The TCP/IP started task must be running.
- The IBM IMS TCP/IP sockets interface, including IMS configuration files, programs, maps, and transactions, must be installed.
- Both the install PC client and the target host (OS/390 / z/OS) should have FTP enabled (optional).
 - CONNX fully supports the IMS add-on product *IMS Library Integrity Utilities for z/OS* to make the importing of metadata easier. If you wish to use this feature, verify that the utilities are installed on your system. Note: this feature is optional.
- Verify that each PCB within the PSB is either labeled or is identified with a PCB name. The PCB is accessed by name and not by relative PCB number within the PSB. The PSB can be named in two ways:
 1. Add a label to each PCB macro line (columns 1-8). We recommend you use this method.
 2. Add a PCBNAME= parameter to the PCB macro.
- Regenerate each of the modified PSBs so the PSBLIB reflects the identifier for each PCB.
- The DLI interface does not require an IMS Control Region but will run concurrently with an IMS control region if one exists.
- After installation, you may need to add DD statements for your IMS data sets into the CNXDLI member.

Verifying TCP/IP Connectivity

Before you install CONNX on your IMS server, we recommend that you verify the TCP/IP connectivity to your OS/390 z/OS system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

ping hostname

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Step 5 - Install IBM Mainframe-Compatible Server

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 10ms, Average = 2ms

Installing CONNX components on the target IMS system

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears.

2. Select the OS/390 / z/OS tab in the CONNX Server Component Installation dialog box. Select the IMS option in the **Product** area. Select the Started Task option in the **Server Type** area.
3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your OS/390 / z/OS server in the **Server** text box, a TSO user ID in the **User ID** text box and its password in the **Password** text boxes.
4. In the **CONNX Server Data Set** area, enter the data set name of the CONNX server component location (where CONNX for IMS will be installed on the mainframe) in the **CONNX DSN HQL** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text box. If SMS is installed on the CONNX server system, select the **SMS** check box.
5. In the **Load Library Parameters** area, enter the dataset name of the link step output files in the **Load Library DSN** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text box.

Enter the dataset name of the IMS High Level Qualifier in the **IMS HQL** text box and the dataset name of

the library where you will find the *IMS Library Integrity Utilities for z/OS* in the **IMS Utility Location** text box. **Note:** The use of these utilities allows CONNX to automatically import IMS metadata by reading the content of the IMS PSBs and DBDs. If these utilities are not present on the system, leave this field blank.

In this case, you will need to provide an index text specification file along with a COBOL copybook in order to import metadata into the CDD. Please see the CONNX User Guide for more information on importing IMS metadata into a CDD.

NOTE: The **Load Library DSN** name is automatically generated when you enter the data set name of the CONNX server component location in the **CONNX Server Data Set** area.

6. In the **FTP Mode area**, enter the FTP mode that is appropriate for your FTP server. If the host cannot be accessed via FTP, select Manual Copy. Manual Copy will create a local copy of the files that need to be transferred along with a text file with instructions.

7. In the **IMS Interface** area, select the interface you are using to access IMS.

8. Click the **Install** button. This action starts an FTP session to copy the selected components from the client PC to the target host. It may take several minutes for the file transfer process to complete.

9. Click **Close**.

Required and Optional Entry Fields for IMS Server Setup

Group Box	Term	Description
Login Information	Server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.123
	User ID	Your 1-8 character TSO logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character TSO password. All characters convert from lowercase to uppercase.
Server Type	CICS	
	Started Task	Select Started Task.
Product	ADABAS	
	IMS	Select IMS
	VSAM	
Load Library Parameters	Load Library DSN	Link into Target Load Library. A fully-qualified partitioned data set name into which the CONNX Started Task object code is linked. This can be a new partitioned data set, or an existing PDS. In the current example, the target load library is CONNX.STASK.LOAD.
	SMS	Select this if your system uses SMS. If selected, do not specify any VOLSER and UNIT.
	VOLSER	The DASD volume serial on which the new CONNX load library is created. Specify a VOLSER and a UNIT to create the PDS on a specific volume.
	UNIT	The DASD device type (3380, 3390, etc.) of the above VOLSER.
	IMS HLQ	A one- or multi-part high-level data set prefix, which is used to create the CONNX IMS data sets on

		the target host.
	IMS Utility HLQ	A one- or multi-part high-level data set prefix, which is used by the <i>IMS Library Integrity Utilities for z/OS</i> . The output from these utilities are used to build the CDD. Note: This is an optional feature. If the <i>IMS Library Integrity Utilities for z/OS</i> are not installed, this parameter is not used.
CONNX Server Data Set	CONNX DSN HLQ	A one- or multi-part high-level data set prefix, which is used to create the CONNX installation sequential and partitioned data sets on the target host. In the current example, the installation prefix is CONNX.STASK. See the DSN Prefix Data Sets table for a description of the data sets created with this prefix.
	VOLSER	Optional. The DASD volume serial on which the CONNX.STASK sequential and partitioned data sets are created.
	UNIT	The DASD device type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
IMS Interface	ODBA	Use the IMS on-line interface
	DLI	Use the IMS batch interface
TCP/IP	HLQ	A one- or multi-part high-level data set prefix, which is used to create the file where the TCP/IP stack is stored.
	Library	The name of the TCP/IP Library.

IMS DSN Prefix Data Sets

Data Set	Description
CONNX.STASK.SYSIN	Sequential input data set for CONNX IMS.
CONNX.STASK.SYSOUT	Sequential output data set for CONNX IMS.
CONNX.STASK.SYSPRINT	Sequential output data set for CONNX IMS.

Members of PDS CONNX.STASK.CNTL

Member	Purpose
INST	The CONNX Started Task installation job built by the CONNX installation program
REMOVE	CONNX Started Task uninstallation job
CNXIDC	JCL procedure to execute IDCAMS to delete/define/load CONNX IMS / PDS components
CNXLINKB	JCL procedure to execute the IBM linkage editor against CONNX prelinked object code and create the CONNX Started Task load modules
CNXLOAD	JCL procedure to create a new load library for CONNX Started Task components
CNXPARMS	Sample control records which define SAS/C environment variables for the CONNX Started Task TCP/IP Listener and Server programs (CNXRUNI and CNXIMD0B)
CNXIMS	JCL procedure to invoke the CONNX Started Task TCP/IP Listener/Server programs to access the IMS subsystem
CNXIMSJ	Execution JCL for procedure CNXIMS
LNKCFGB	Linkage editor control records for load module CNXCFGB
LNKRUNI	Linkage editor control records for load module CNXRUNI
LNKIMSB	Linkage editor control records for load module CNXIMD0B

IMS Installation JCL

After the file transfers complete, review and run the following installation jobs.

1. Review the INST JCL in the CNTL data set and make any modifications necessary to adhere to your environment.
2. Submit the INST JCL and review its output. The steps should normally return condition codes of 4 or less except for data sets which do not exist or are allocated to other jobs or tasks. These steps will return codes of 8.
3. Review the CNXIMSJ JCL and make any modifications necessary to adhere to your environment.
4. Submit the CNXIMSJ JCL. This starts the CONNX listener. You can confirm this is running through the SDSF menu or through the TSO NETSTAT command.

This completes the CONNX for IMS installation.

Uninstalling the CONNX IMS components

The CONNX for IMS TCP/IP Server Setup client application transfers the CONNX components to the target OS/390 / z/OS host using TCP FTP. After the initial FTP step, the CONNX installer submits a batch job using TSO ISPF. The optional installation steps include manual additions of a JCL procedure (CNXIMS) and a REXX exec file (CNXSTOP) to the system PROCLIB and SYSPROC partitioned data sets.

Given a default CONNX installation prefix of CONNX.STASK, the CONNX for IMS Started Task installation JCL partitioned data set is CONNX.STASK.CNTL. Member REMOVE contains batch uninstallation JCL .

To uninstall the CONNX for IMS Started Task components

1. Before submitting the uninstall JCL, find and stop all instances of the CONNX Started Task/Batch Job TCP/IP Listener program CNXRUNI. For example, if batch job CNXIMSJ is executing, stop it.
2. Use the TSO NETSTAT command to list the TCP/IP listen port dedicated to CNXIMSJ, and all active client/server connections:

```
EZZ2587I CNXIMSJ  0009C426 123.123.123.123..1927 123.123.123.124..4397
      Establish
```

```
EZZ2587I CNXIMSJ  0009C424 0.0.0.0..6789          0.0.0.0..0
      Listen
```

```
EZZ2587I CNXIMSJ  0009C42D 123.123.123.123..1930 123.123.123.125..4405
      Establish
```

```
EZZ2587I CNXIMSJ  0009C429 123.123.123.123..1928 123.123.123.126..4401
      Establish
```

```
EZZ2587I CNXIMSJ  0009C42B 123.123.123.123..1929 123.123.123.127..4403
      Establish
```

3. Use the TSO CNXSTOP command, using the CNXRUNPORT specified in the CNXPARMs file, to terminate the CNXRUNI listener program and its active subtasks.
4. Repeat these steps for any other batch jobs/started tasks which are currently executing the CONNX TCP/IP listener program CNXRUNI.
5. After stopping all instances of the CNXIMS Started Task/Batch Jobs, you can submit the JCL in member @REMOVE. The REMOVE job is set up with TYPERUN=HOLD and will need to be released to run after making sure all CONNX.STASK data sets are not being accessed.

The second IDCAMS DELETE step for the CONNX.STASK.LOAD partitioned data set is commented out, in case the CONNX Started Task/Batch job load modules were linked into an existing load library. We recommend that you install CONNX into its own load library.

6. The IDCAMS DELETE steps should return zeroes for all data sets which are not allocated to another batch job, started task, or TSO session. For data sets which do not exist or are allocated to other jobs or tasks, the IDCAMS return code is 8. Any undeleted data sets can be deleted via ISPF or via a second run of the REMOVE JCL after the dataset sharing conflicts have been resolved.

VSAM

CONNX for VSAM - Started Task/Batch

Prerequisites - CONNX for VSAM / QSAM / PDS

CONNX for VSAM / QSAM / PDS requires that:

- The target system be set up to support TCP/IP communications.
- The TCP/IP Started Task must be running.
- Read/write FTP access be configured and available on the target system.
- Both the installation PC client and the target host (OS/390 z/OS) must have FTP enabled.

CONNX for VSAM / QSAM / PDS can access any VSAM, QSAM, or partitioned data set defined on the target OS/390 / z/OS system, as long as the host user ID has the necessary privileges as defined via IBM RACF (Resource Access CONTROL Facility) or a RACF-compatible product such as eTrust CA-Top Secret Security for z/OS & OS/390®. COBOL copybooks for the target files must reside in partitioned data sets defined for FTP read access.

Verifying TCP/IP Connectivity

Before you install CONNX on your VSAM server, we recommend that you verify the TCP/IP connectivity to your OS/390 z/OS system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

ping hostname

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Part 1: Installation Procedures

To install the CONNX components on the target z/OS system

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears.

2. Select the OS/390 / z/OS tab in the CONNX Server Component Installation dialog box. Select the VSAM option in the **Product** area. Select the Started Task option in the **Server Type** area.

Note: Additional information about the information in these dialog boxes can be found in the **Required and Optional Entry Fields for VSAM TCP/IP Server Setup** table.

3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your OS/390 / z/OS server in the **Server** text box, a TSO user ID in the **User ID** text box and its password in the **Password** text boxes.
4. In the **TCP/IP area**, enter the dataset name of the TCP/IP stack in the **HLQ** text box and the TCP/IP library name in the **Library** text box. The TCP/IP library is the location of the object **EZALBADM**. The default library name on z/OS versions 1.5 and newer is SEZALNK2. On older systems, the default library name is SEZALINK.
5. In the **CONNX Server Data Set** area, enter the data set name of the CONNX server component location (where CONNX for VSAM will be installed on the mainframe) in the **CONNX DSN HQL** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text box. If SMS is installed

on the CONNX server system, select the **SMS** check box.

6. In the **Load Library Parameters** area, enter the dataset name of the link step output files in the **Load Library DSN** text box. If SMS is enabled, select the SMS checkbox otherwise enter the VOLSER in the **VOLSER** text box, and the DASD device type in the **UNIT** text box.

7. In the **VSAM Samples** area, select the **Create** check box if you want to create VSAM sample files. Enter the data set name of the CONNX sample file location in **Database HLQ** and its VOLSER in the **VOLSER** text boxes. If the data set prefix is CONNX.STASK, this will create the VSAM, QSAM, and partitioned data sets listed in [VSAM Objects for CONNX.STASK.SAMPLES](#) and load them with sample data.

If you do not want sample files, clear the **Create** check box.

8. In the **FTP Mode area**, enter the FTP mode that is appropriate for your FTP server. If the host cannot be accessed via FTP, select Manual Copy. Manual Copy will create a local copy of the files that need to be transferred along with a text file with instructions.

9. Click the **Install** button. This action starts an FTP session and copies the selected components from the client PC to the target host.

10. Allow several minutes for the file transfer process between the CONNX administrator computer directory C:\CONN32\VSAM and the target host.

11. Click the **Close** button.

Required and Optional Entry Fields for VSAM TCP/IP Server Setup

Group Box	Term	Description
Login	VSAM Server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.123
	User ID	Your 1-8 character TSO logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character TSO password. All characters convert from lowercase to uppercase.
Server Type	CICS	
	Started Task	Select Started Task.
VSAM Operating System		Select OS/390.
Started Task Parameters	Link into Target Load Library	Link into Target Load Library Required for both the CICS and Started Task MVS / OS/390 CONNX installation options. A fully-qualified partitioned data set name into which the CONNX Started Task C++ object code is linked. This can be a new partitioned data set, or an existing PDS. In the current example, the target load library is CONNX.STASK.LOAD.
	VOLSER	The DASD volume serial on which the new CONNX load library is created. Specify a VOLSER and a UNIT to create the PDS on a specific volume, or leave both fields blank to accept the system defaults. If linking into an existing PDS, leave the VOLSER and UNIT text boxes blank.
	UNIT	The DASD device type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
CONNX Server Data Set	CONNX DSN Prefix	CONNX DSN Prefix A one- or multi-part high-level data

		set prefix, which is used to create the CONNX installation sequential and partitioned data sets on the target host. In the current example, the installation prefix is CONNX.STASK. See the DSN Prefix Data Sets table for a description of the data sets created with this prefix.
	VOLSER	Optional: The DASD volume serial on which the CONNX.STASK sequential and partitioned data sets are created.
	UNIT	Optional. The DASD device type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
VSAM Samples	VSAM Samples Prefix	Required if the Create check box is checked. A one- or multi-part high-level data set name prefix used to create the CONNX sample VSAM, QSAM, and partitioned data sets. Given the default data set prefix of CONNX.STASK, the VSAM, QSAM, and partitioned data sets listed in VSAM Objects for CONNX.STASK.SAMPLES are created and loaded with sample data.
	VOLSER	Required if the Create check box is checked. The DASD volume serial on which the CONNX.STASK.SAMPLES VSAM, QSAM, and sample partitioned data sets are created.

DSN Prefix Data Sets

Data Set	Description
CONNX.STASK.CNTL	Installation JCL and control records partitioned data set
CONNX.STASK.COPYBOOK	Sample COBOL copybook PDS
CONNX.STASK.CUSTOMER.S	Sequential input data set used to load the sample CUSTOMER VSAM files
CONNX.STASK.EQUIPMNT.S	Sequential input data set used to load the sample EQUIPMENT VSAM files
CONNX.STASK.LOAD	CONNX load library partitioned data set for Started Task TCP/IP listener/server and configuration utility programs.
CONNX.STASK.OBJECT	CONNX SAS/C 7.00 pre-linked object code PDS for the CONNX Started Task programs
CONNX.STASK.ORDER.S	Sequential input data set used to load the sample ORDER VSAM files
CONNX.STASK.PRODGRP.S	Sequential input data set used to load the sample PRODGRP (Product Group) VSAM files
CONNX.STASK.PRODUCT.S	Sequential input data set used to load the sample PRODUCT VSAM files
CONNX.STASK.STDERR	Sequential output data set for SAS/C library run-time error messages
CONNX.STASK.STDOUT	Sequential output data set for CONNX debug error messages
CONNX.STASK.STGRPT	Sequential output data set for SAS/C virtual memory storage reports

VSAM Objects for CONNX.STASK.SAMPLES

CONNX Sample VSAM / QSAM/ PDS Name	Description
CONNX.STASK.SAMPLES.CUSTOMER	Customer file KSDS.
CONNX.STASK.SAMPLES.CUSTOMER.E	Customer file ESDS
CONNX.STASK.SAMPLES.CUSTOMER.PDS	Customer-partitioned data set
CONNX.STASK.SAMPLES.CUSTOMER.PS	Customer physical sequential file
CONNX.STASK.SAMPLES.CUSTOMER.R	Customer file RRDS
CONNX.STASK.SAMPLES.EQALT01	Equipment KSDS AIX #1
CONNX.STASK.SAMPLES.EQALT02	Equipment KSDS AIX #2
CONNX.STASK.SAMPLES.EQPATH01	Equipment KSDS path #1
CONNX.STASK.SAMPLES.EQPATH02	Equipment KSDS path #2
CONNX.STASK.SAMPLES.EQUIPMNT	Equipment file KSDS
CONNX.STASK.SAMPLES.EQUIPMNT.E	Equipment file ESDS
CONNX.STASK.SAMPLES.EQUIPMNT.PDS	Equipment-partitioned data set
CONNX.STASK.SAMPLES.EQUIPMNT.PS	Equipment physical sequential file
CONNX.STASK.SAMPLES.EQUIPMNT.R	Equipment file RRDS
CONNX.STASK.SAMPLES.ORDALT01	Order KSDS AIX #1
CONNX.STASK.SAMPLES.ORDALT02	Order KSDS AIX #2
CONNX.STASK.SAMPLES.ORDER	Order file KSDS
CONNX.STASK.SAMPLES.ORDER.E	Order file ESDS
CONNX.STASK.SAMPLES.ORDER.PDS	Order-partitioned data set
CONNX.STASK.SAMPLES.ORDER.PS	Order physical sequential file

CONNX.STASK.SAMPLES.ORDER.R	Order file RRDS
CONNX.STASK.SAMPLES.ORDPTH01	Order KSDS path #1
CONNX.STASK.SAMPLES.ORDPTH02	Order KSDS path #2
CONNX.STASK.SAMPLES.PRODGRP	Product group KSDS
CONNX.STASK.SAMPLES.PRODGRP.E	Product group ESDS
CONNX.STASK.SAMPLES.PRODGRP.PDS	Product group partitioned data set
CONNX.STASK.SAMPLES.PRODGRP.PS	Product group physical sequential file
CONNX.STASK.SAMPLES.PRODGRP.R	Product group RRDS
CONNX.STASK.SAMPLES.PRODUCT	Product KSDS
CONNX.STASK.SAMPLES.PRODUCT.E	Product ESDS
CONNX.STASK.SAMPLES.PRODUCT.PDS	Product group partitioned data set
CONNX.STASK.SAMPLES.PRODUCT.PS	Product physical sequential file
CONNX.STASK.SAMPLES.PRODUCT.R	Product RRDS

Started Task Installation

Introduction to Started Task installation procedures

Once the ftp successfully completes, you are ready to install the selected CONNX components. To begin this process, log on to TSO on the target system. Given the CONNX Server Data set prefix above = CONNX.STASK, the ftp process creates a JCL/control record partitioned data set with the name and members listed in the following table:

Default Partitioned Data Set Name = CONNX.STASK.CNTL

Member	Purpose
INST	The CONNX Started Task installation job built by the CONNX installation program.
REMOVE	CONNX Started Task uninstallation job + delete steps for sample files
REMOV01	CONNX Started Task uninstallation job - delete steps for sample files
CICSALOC	JCL procedure to allocate the SAS/C STDERR, STDOUT, and STGRPT output data sets for SAS/C runtime- and CONNX debug- error messages and SAS/C virtual memory usage reporting
CNXIDC	JCL procedure to execute IDCAMS to delete/define/load CONNX VSAM / QSAM / PDS components
CNXLINKB	JCL procedure to execute the IBM linkage editor against CONNX prelinked SAS/C 7.00 object code and create the CONNX Started Task load modules
CNXLOAD	JCL procedure to create a new load library for CONNX Started Task components
CNXPARMS	Sample control records which define SAS/C environment variables for the CONNX Started Task TCP/IP Listener and Server programs (CNXRUNB and CNXVSD0B)
CNXVSAM	JCL procedure to invoke the CONNX Started Task TCP/IP Listener/Server programs to access the CONNX VSAM and QSAM sample files
CNXVSAMJ	Execution JCL for procedure CNXVSAM
CNXVSCT	JCL procedure to invoke the CONNX Started Task TCP/IP Listener/Server programs to access the CONNX VSAM / QSAM / PDS sample files and the SCT master files
CNXVSCTJ	Execution JCL for procedure CNXVSCT
CNXVSCT1	Same as procedure CNXVSCT, minus the CONNX sample DDNAMEs
CNXVSJ01	Execution JCL for procedure CNXVSM01
CNXVSJ02	Execution JCL for procedure CNXVSCT1

CNXVSM01	Same as procedure CNXVSAM, minus the CONNX sample DDNAMEs
IDCCNX0	IDCAMS control records to delete/define the CONNX TCP/IP server VSAM configuration file
IDCCUST0	IDCAMS control records to delete/define/load the CONNX sample KSDS, ESDS, and RRDS customer files
IDCEQU00	IDCAMS control records to delete/define/load the CONNX sample KSDS, ESDS, and RRDS equipment files
IDCEQU01	IDCAMS control records to define the CONNX sample KSDS equipment file alternate index #1
IDCEQU02	IDCAMS control records to define the CONNX sample KSDS equipment file alternate index #2
IDCEQU03	IDCAMS control records to build the CONNX sample KSDS equipment file alternate index #1
IDCEQU04	IDCAMS control records to build the CONNX sample KSDS equipment file alternate index #2
IDCORD00	IDCAMS control records to delete/define/load the CONNX sample KSDS, ESDS, and RRDS order files
IDCORD01	IDCAMS control records to define the CONNX sample KSDS order file alternate index #1
IDCORD02	IDCAMS control records to define the CONNX sample KSDS order file alternate index #2
IDCORD03	IDCAMS control records to build the CONNX sample KSDS order file alternate index #1
IDCORD04	IDCAMS control records to build the CONNX sample KSDS order file alternate index #4
IDCPGRP0	IDCAMS control records to delete/define/load the CONNX sample KSDS, ESDS, and RRDS product group files
IDCPROD0	IDCAMS control records to delete/define/load the CONNX sample KSDS, ESDS, and RRDS product files
LNKCFGB	Linkage editor control records for load module CNXCFGB
LNKRUNB	Linkage editor control records for load module CNXRUNB
LNKVSAMB	Linkage editor control records for load module CNXVSD0B

Once the members have been verified, the next step is to install the server component

To install the server component, edit the **INST JCL** and make any necessary modifications to the job card. When you have completed any JCL modifications, submit the job. This will create the

load modules in the load library specified as well as create sample files if needed.

The return code from INST should be ≤ 4 . If it is greater than 4, please review the job output and correct any errors.

After successfully running the INST JCL, review the CNXPARMs file in the CNTL directory. Please take note of the following values:

- CNXRUNPORT defaults to 6500. This is the port the CONNX listener will listen on. If this port is not available on your system, please change the value to a port that is available.
- CNXNOPREAUTHORIZE defaults to 0. A value of 0 means the CONNX server will validate the User ID and password. If password validation is being used, the CONNX load library along with all the libraries in the STEPLIB must be APF authorized. If you do not wish to use User ID and password authentication, set this value to 1.
- ALLOWMIXEDPWD defaults to 1. If your z/OS system does not have mixed case password support enabled, set this value to 0.

Once the parameters are set to your liking in CNXPARMs, edit the **CNXVSAMJ** JCL and make any necessary adjustments to the job card.

Submit CNXVSAMJ to start the CONNX listener.

For more information on the above steps, see the section "[To verify Installation of CONNX Start Task components](#)" in the next chapter.

Part 2: Installation Verification of CONNX Started Task components

Introduction to installation verification

After the CONNX installation runs to completion, the first verification step is to check that all job steps completed with condition codes less than or equal to four. A successful installation of the CONNX Started Task components creates 76 data sets with the default user-modifiable prefix of CONNX.STASK. The CONNX VSAM / QSAM / PDS test data sets are created with a default user-modifiable prefix of CONNX.STASK.SAMPLES. These test data sets are for demonstration purposes only, and can be safely deleted. The minimal complement of data sets necessary to run CONNX for VSAM / QSAM / PDS, given the default data set prefix of CONNX.STASK, is as follows:

CONNX.STASK.CNTL

CONNX.STASK.OBJECT

CONNX.STASK.LOAD

The following sections enumerate the manual steps necessary to complete the installation of the started task components of CONNX for VSAM / QSAM / PDS.

To verify installation of CONNX Started Task components

1. Given the default installation prefix of CONNX.STASK, the following data sets are created by a successful complete installation job:

```

CONNX.STASK.CNTL
CONNX.STASK.COPYBOOK
CONNX.STASK.CUSTOMER.S
CONNX.STASK.EQUIPMNT.S
CONNX.STASK.LOAD
CONNX.STASK.OBJECT
CONNX.STASK.ORDER.S
CONNX.STASK.PRODGRP.S
CONNX.STASK.PRODUCT.S
CONNX.STASK.SAMPLES.CUSTOMER
CONNX.STASK.SAMPLES.CUSTOMER.D
CONNX.STASK.SAMPLES.CUSTOMER.E
CONNX.STASK.SAMPLES.CUSTOMER.E.D
CONNX.STASK.SAMPLES.CUSTOMER.I
CONNX.STASK.SAMPLES.CUSTOMER.PDS
CONNX.STASK.SAMPLES.CUSTOMER.PS
CONNX.STASK.SAMPLES.CUSTOMER.R
CONNX.STASK.SAMPLES.CUSTOMER.R.D
CONNX.STASK.SAMPLES.EQALT01
CONNX.STASK.SAMPLES.EQALT01.D
CONNX.STASK.SAMPLES.EQALT01.I
CONNX.STASK.SAMPLES.EQALT02
CONNX.STASK.SAMPLES.EQALT02.D
CONNX.STASK.SAMPLES.EQALT02.I
CONNX.STASK.SAMPLES.EQPATH01
CONNX.STASK.SAMPLES.EQPATH02
CONNX.STASK.SAMPLES.EQUIPMNT
CONNX.STASK.SAMPLES.EQUIPMNT.D
CONNX.STASK.SAMPLES.EQUIPMNT.E
  CONNX.STASK.SAMPLES.EQUIPMNT.E.D
  CONNX.STASK.SAMPLES.EQUIPMNT.I
  CONNX.STASK.SAMPLES.EQUIPMNT.PDS
  CONNX.STASK.SAMPLES.EQUIPMNT.PS
  CONNX.STASK.SAMPLES.EQUIPMNT.R
  CONNX.STASK.SAMPLES.EQUIPMNT.R.D
CONNX.STASK.SAMPLES.ORDALT01
CONNX.STASK.SAMPLES.ORDALT01.D
CONNX.STASK.SAMPLES.ORDALT01.I
CONNX.STASK.SAMPLES.ORDALT02
CONNX.STASK.SAMPLES.ORDALT02.D
CONNX.STASK.SAMPLES.ORDALT02.I
CONNX.STASK.SAMPLES.ORDER
CONNX.STASK.SAMPLES.ORDER.D
CONNX.STASK.SAMPLES.ORDER.E

```

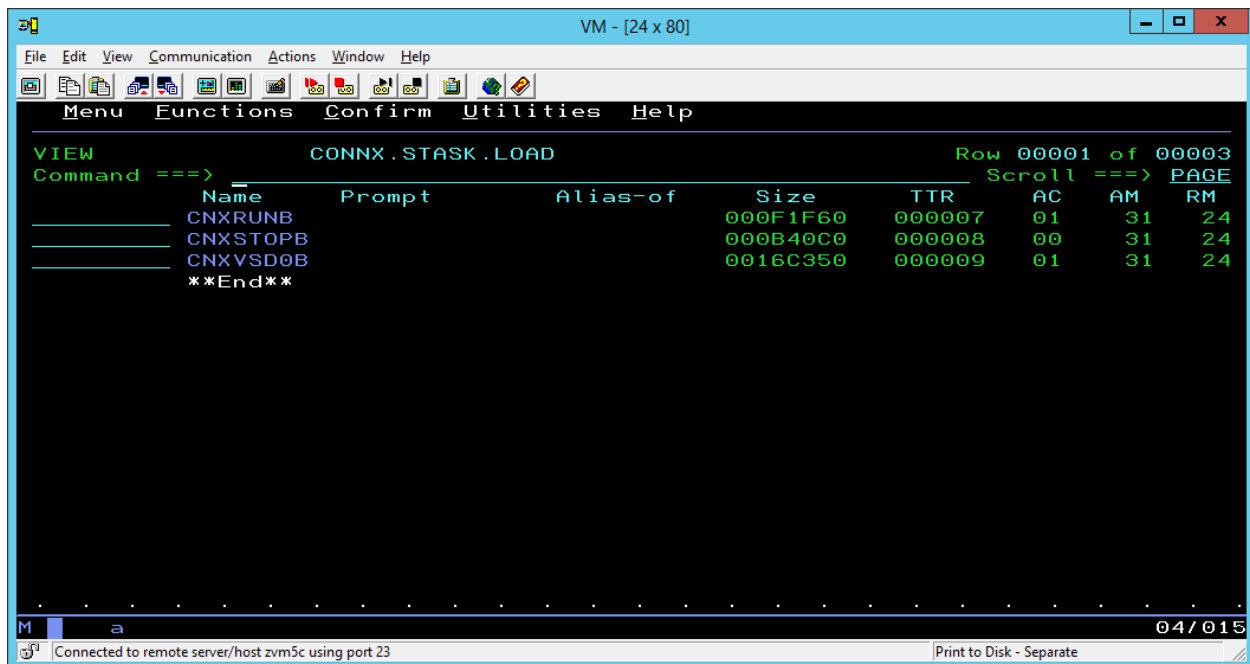


```

CONNX.STASK.SAMPLES.ORDER.E.D
CONNX.STASK.SAMPLES.ORDER.I
CONNX.STASK.SAMPLES.ORDER.PDS
CONNX.STASK.SAMPLES.ORDER.PS
CONNX.STASK.SAMPLES.ORDER.R
CONNX.STASK.SAMPLES.ORDER.R.D
CONNX.STASK.SAMPLES.ORDPTH01
CONNX.STASK.SAMPLES.ORDPTH02
CONNX.STASK.SAMPLES.PRODGRP
CONNX.STASK.SAMPLES.PRODGRP.D
CONNX.STASK.SAMPLES.PRODGRP.E
CONNX.STASK.SAMPLES.PRODGRP.E.D
CONNX.STASK.SAMPLES.PRODGRP.I
CONNX.STASK.SAMPLES.PRODGRP.PDS
CONNX.STASK.SAMPLES.PRODGRP.PS
CONNX.STASK.SAMPLES.PRODGRP.R
CONNX.STASK.SAMPLES.PRODGRP.R.D
CONNX.STASK.SAMPLES.PRODUCT
CONNX.STASK.SAMPLES.PRODUCT.D
CONNX.STASK.SAMPLES.PRODUCT.E
CONNX.STASK.SAMPLES.PRODUCT.E.D
CONNX.STASK.SAMPLES.PRODUCT.I
CONNX.STASK.SAMPLES.PRODUCT.PDS
CONNX.STASK.SAMPLES.PRODUCT.PS
CONNX.STASK.SAMPLES.PRODUCT.R
CONNX.STASK.SAMPLES.PRODUCT.R.D

```

2. Browse the CONNX.STASK.LOAD PDS to ensure that the following load modules are present and executable:



Note: CNXRUNB and CNXVSD0B are built with AC flag =1, which means that they must reside in an APF-authorized load library. This requirement can be temporarily deferred by changing the CONNX configuration variable CNXNOPREAUTHORIZE to 1 in the CNXPARMS file. By default, the CONNX Started Task TCP/IP Listener program (CNXRUNB) authenticates Userid / Password entries received from the CONNX client layer by calling the IBM Resource Access Control Facility (RACF) VERIFY exit. To bypass this logic and the APF-authorized load library requirement, define the following CONNX environment variable / value pair:

=CNXNOPREAUTHORIZE=1

Refer to member CNXPARMS of CONNX.STASK.CNTL for the default CONNX parameter settings. It is strongly recommended that the default CNXNOPREAUTHORIZE value of 0 not be overridden, unless the target system does not implement RACF or a similar RACF-compatible security product.

3. Browse CONNX.STASK.CNTL and find members CNXVSAM, CNXVSM01, CNXVSAMJ, and CNXVSJ01. Members CNXVSAM and CNXVSM01 define JCL procedures which invoke the CONNX Started Task TCP/IP Listener/Server programs. CNXVSAM includes parameters and DDNAMEs for the CONNX sample VSAM / QSAM / PDS files. The batch job execution JCL for these procedures is in members CNXVSAMJ and CNXVSJ01, respectively. Review the following procedure parameters:

CNXHLQ	High-level qualifier for CONNX JCL and load library partitioned data sets and CONNX VSAM configuration file
CNXSAMP	High-level qualifier for CONNX sample QSAM, PDS, and VSAM KSDS, ESDS, and RRDS files
CNXPARMS	Member name for CONNX environment variable control records
TCPHLQ	High-level qualifier for TCP/IP load libraries
REG	Job step region parameter

The parameters are pre-set with values entered from the CONNX VSAM TCP/IP Server Setup program. Members CNXVSAMJ and CNXVSJ01 contain commented-out overrides for these parameters.

4. Edit member CNXPARMS of CONNX.STASK.CNTL and review the pre-set CONNX environment variable entries:

Variable	Supported Values
DEBUG=0	H 0 or 1. 1 enables CONNX debug tracing message logic. Default = 0, which disables the tracing logic. The messages are written to DDNAME STDOUT.
CNXRUNPORT=6500	The TCP/IP port on which the CONNX TCP/IP Listener program accepts messages from the CONNX Windows client interfaces (ODBC, JDBC, OLE-DB, and .NET). Default = 6500.
CNXNOPREAUTHORIZE=0	Enables / disables the CONNX userid / password RACF verification logic. 1 = disable; 0 = enable. Default = 0 (enable verification)
CNXCONNECTBACK=1	Enables / Disables TCP/IP socket sharing in the CONNX for VSAM TCP/IP Listener connection logic. Default = 1 = enable; the optimal setting for the CONNX for VSAM Started Task product is 1.

5. Change:

FROM	TO
CNXNOPREAUTHORIZE=0	CNXNOPREAUTHORIZE=1
DEBUG=0	DEBUG=1

6. Changing CNXNOPREAUTHORIZE to 1 disables the CONNX userid / password verification logic and thus permits the CONNX for VSAM / QSAM / PDS Started Task programs to execute in a non-APF (Authorized Program Facility) load library. Setting this variable to 0 requires that the CONNX.STASK.LOAD load library be added to the APF load library list. For more information related to this environment variable, refer to the topic [To test the CONNX for VSAM / QSAM / PDS Started Task / Batch Job Userid/Password Verification](#).

7. If TCP/IP port 6500 is in use by another application, change the entry for:

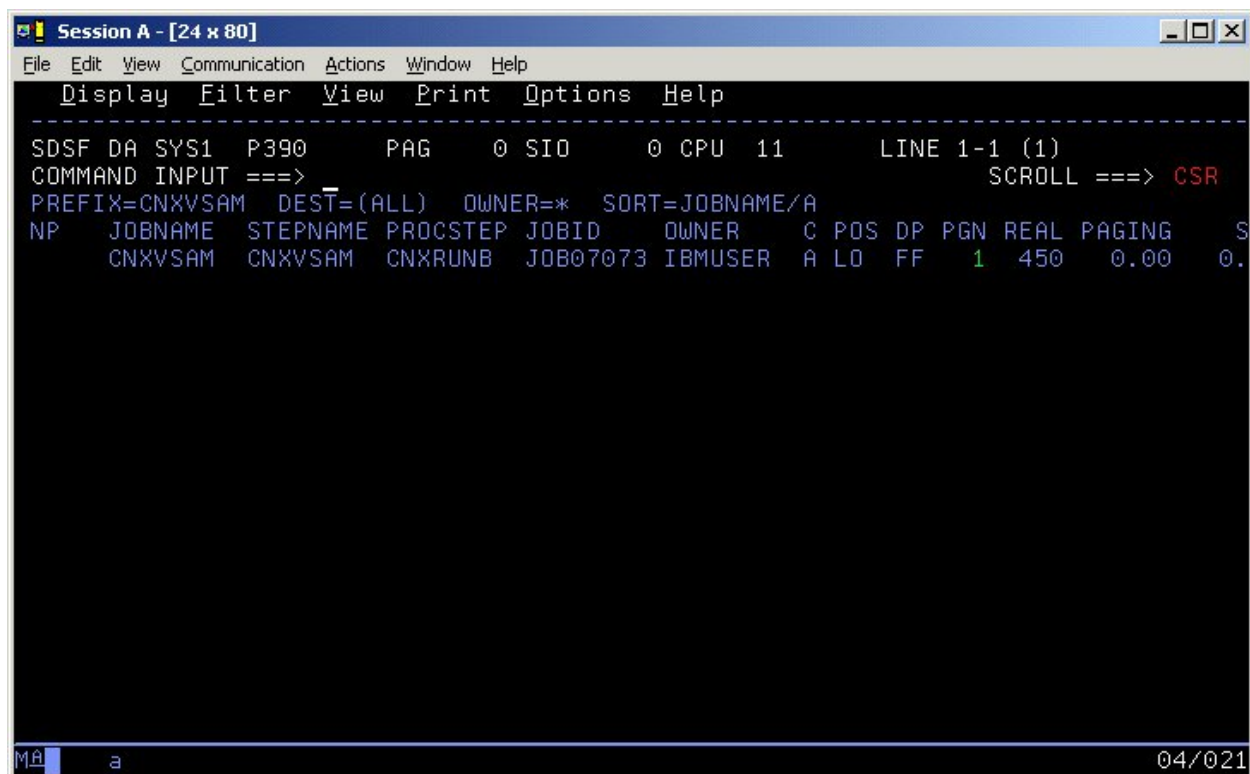
=CNXRUNPORT=6500

to an unused port.

8. Edit member CNXVSAMJ, change the job card parameters to match your installation standards, and submit the job to a batch execution queue. The CONNX TCP/IP Listener program (CNXRUNB) starts execution and attempts to listen on the TCP/IP port defined by the CNXRUNPORT environment variable. The following CONNX debug tracing messages, which are by default written to the CNXVSAM job output queue, show a successful attempt to listen on port 6710:

```
Tues Apr 2 17:59:21 2003
Listening to network port 6710
Tue Apr 2 17:59:26 2003
Binding to port 6710
Tue Apr 2 17:59:26 2003
NET TCP Listening on (0.0.0.0) Port(6710)
```

9. After these messages are written to the job log, the CNXVSAM job waits for a connect request from an ODBC / JDBC / OLE DB / .NET application. An SDSF display of an idle CONNX VSAM TCP/IP batch server job follows:



The screenshot shows a terminal window with a blue title bar and a menu bar. The main area is black with white and red text. The text displayed is as follows:

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390      PAG    0 SIO      0 CPU  11      LINE 1-1 (1)
COMMAND INPUT ==>          SCROLL ==> CSR
PREFIX=CNXVSAM  DEST=(ALL) OWNER=*  SORT=JOBNAME/A
NP  JOBNAME  STEPNAME  PROCSTEP  JOBID    OWNER    C POS  DP  PGN  REAL  PAGING  S
    CNXVSAM  CNXVSAM   CNXRUNB   JOB07073  IBMUSER  A LO   FF   1   450   0.00   0.
```

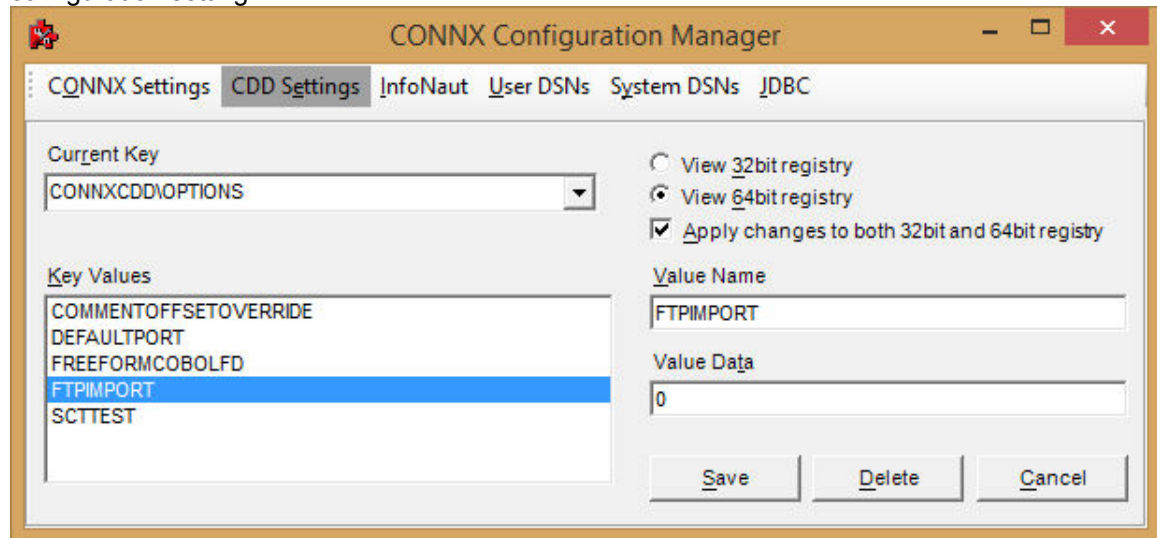
At the bottom left, there is a small box containing 'Mā' and a cursor. At the bottom right, the text '04/021' is displayed.

To import metadata from COBOL copybooks

After the CONNX TCP/IP listener program is started as a batch job, the next step is to use the CONNX Data Dictionary Manager program to import metadata from COBOL copybooks into a CONNX Data Dictionary file. The following steps show how to import metadata for the CONNX sample VSAM customer KSDS (Key Sequenced Data Set) file:

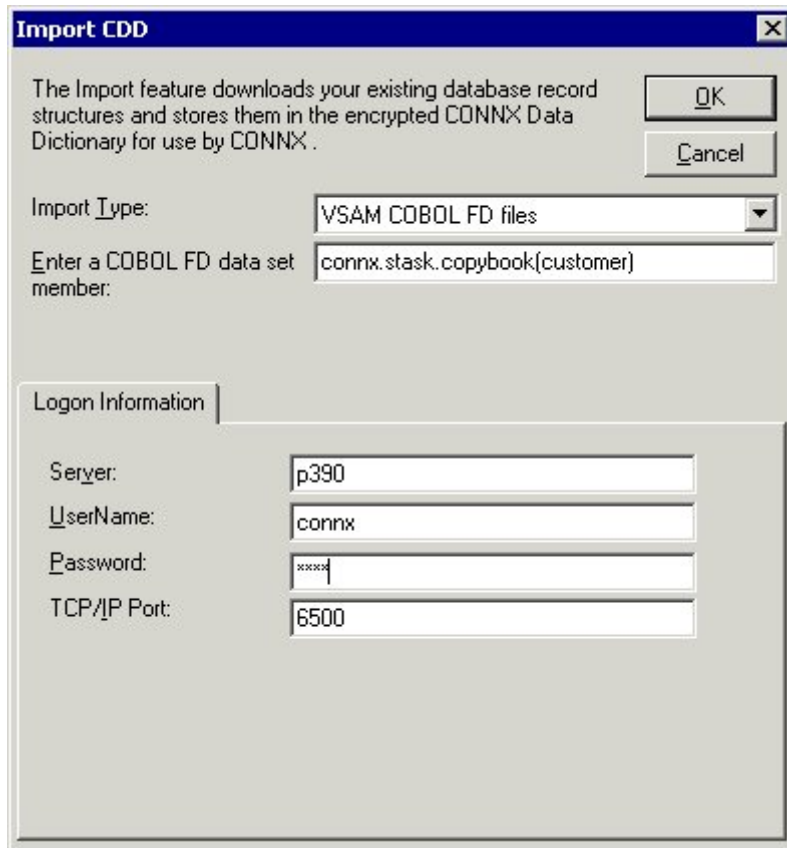
1. Click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX Data Dictionary**. The CONNX Data Dictionary Manager appears.
2. Click the **Import** button. The **Import CDD** dialog box appears.
3. Select **VSAM COBOL FD files** from the **Import Type** list box.
4. In the **Enter a COBOL FD data set member** text box, enter **CONNX.STASK.COPYBOOK (CUSTOMER)**.

By default CONNX will use FTP to import the copybook. To import without FTP add the following configuration setting:



With FTPIMPORT set to zero, the copybook must reside on the PC where the import is being performed (in ascii). Specify the path to the copybook. Wildcard imports are not permitted in this mode.

5. In the **Server** text box, enter the symbolic or dotted numeric TCP/IP address of the target host, and then enter your TSO user name and password. Enter the TCP/IP port number defined by the =CNXRUNPORT environment variable entry in the CNXPARMS member of the CONNX.STASK.CNTL partitioned data set as discussed in [To verify installation of CONNX Started Task components](#).
6. Select a database container in the destination database list box. For more information, see Adding a Database Connection in the CONNX User Reference Guide, available online, on the CONNX CD-ROM, and in hard copy.
7. Click the **OK** button.



Import CDD

The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX.

OK Cancel

Import Type: VSAM COBOL FD files

Enter a COBOL FD data set member: connx.stask.copybook(customer)

Logon Information

Server: p390

UserName: connx

Password: xxxx

TCP/IP Port: 6500

8. Select the **CUSTOMERS_VSAM** table. The **Table Properties** tab appears with the **File Name** text box set to UNKNOWN.FILE.

CONNX - CONNX Data Dictionary Manager

File Edit Security Tools View Help

CONNX Views
 - p390 (VSAM)
 CUSTOMERS_VSAM

Add...
 Rename...
 Delete
 Import...

Table Properties | Table Columns | Table Indexes | Table Security

CICS File Name: UNKNOWN.FILE

Comment:

SQL View Clause:

Record Length: 111

☐ Show CNXROWNUMBER ☐ Show CRC

☐ Show CNXROWID

☐ Virtual ID & Offset

Schema Date: 2003-04-02 00:00:00.00

Priority Count: 0

Large Rotated Array Information

☐ Rotated Array

Actual Size:

Maximum Size: 0

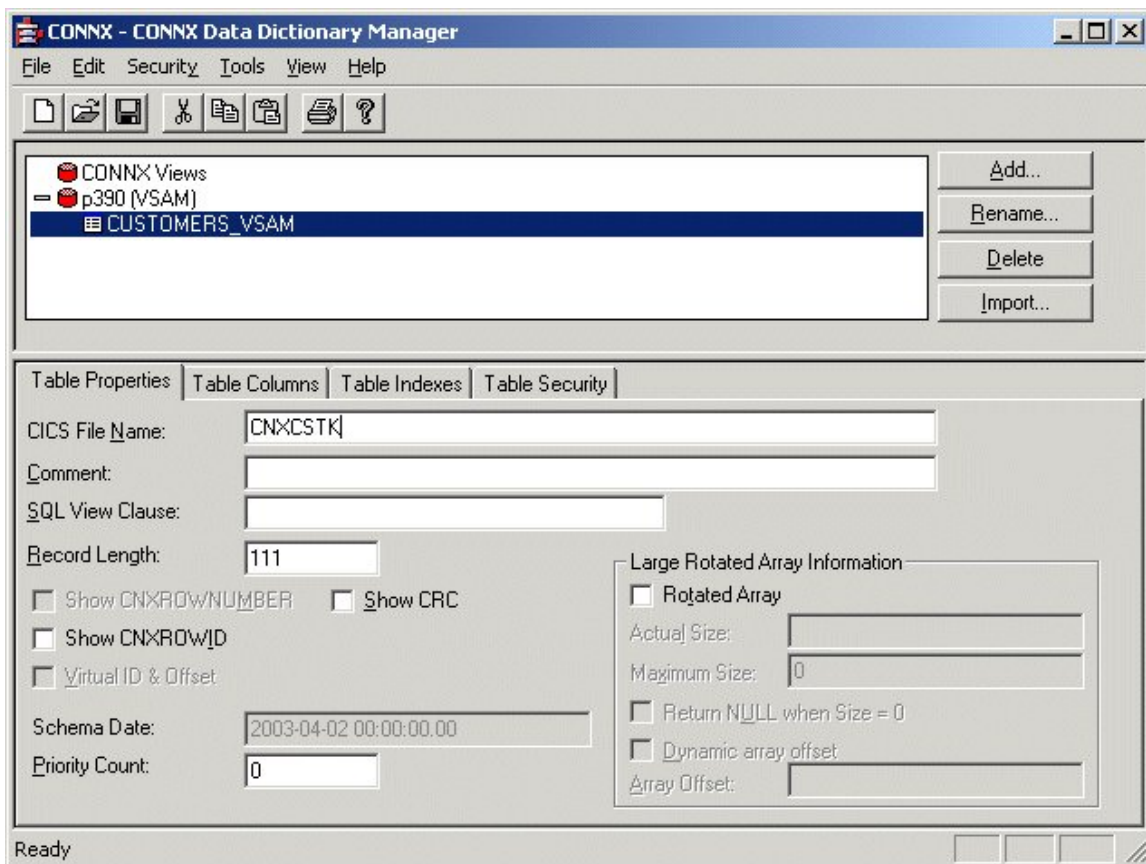
☐ Return NULL when Size = 0

☐ Dynamic array offset

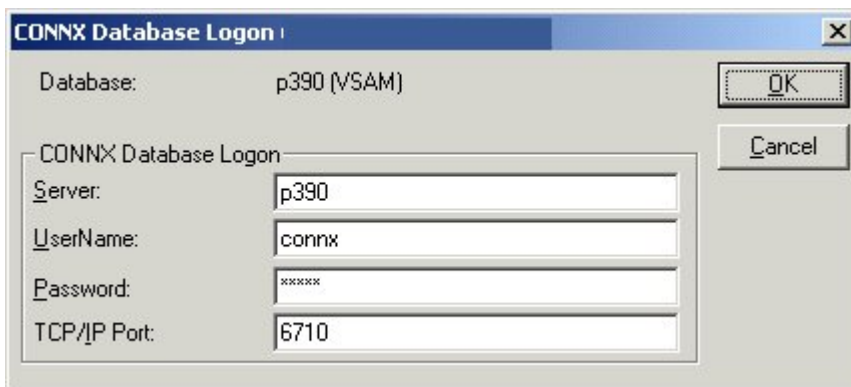
Array Offset:

Ready

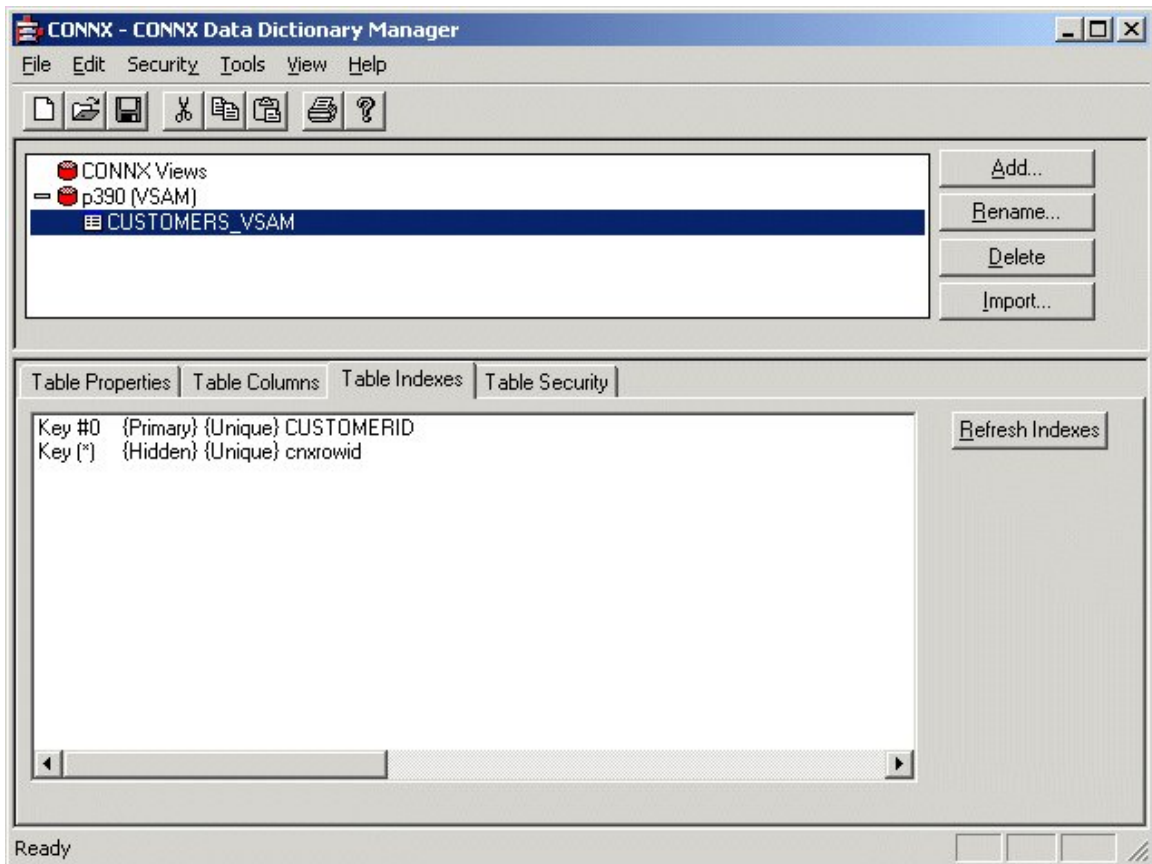
9. For VSAM files, the file layouts are maintained separately from the physical or logical file name (fully qualified JCL DSN or DDNAME, respectively). The CNXVSAM JCL procedure defines standard DDNAMES for the CONNX sample VSAM files. For the CONNX sample VSAM Key-Sequenced Data Set (KSDS) CUSTOMER file, the DDNAME is CNXCSTK. Overtyping **UNKNOWN.FILE** with **CNXCSTK**:



10. Press <Tab> to open the **CONNX Database Logon** dialog box.



11. Enter your TSO user ID and password, and the correct TCP/IP port number, and then click the **OK** button. A successful import of the CNXCSTK file locates a primary key (index) on the CUSTOMERID field, which can be displayed by clicking the **Table Indexes** tab:



12. To import additional CONNX sample KSDS files, refer to CONNX.STASK.COPYBOOK(@INDEX) for sample CONNX COBOL copybooks and VSAM/QSAM JCL DDNAMES, and repeat steps 5 through 10:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

BROWSE CONNX.STASK.COPYBOOK(@INDEX) - 01.0 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR_

***** Top of Data *****
MEMBER      CONNX SAMPLE FILE DESCRIPTION      CICS FILE NAME / JCL DDNAME
CUSTOMER    Customer      VSAM KSDS      CNXCSTK
CUSTOMRE    Customer      VSAM ESDS      CNXCSTE
CUSTOMRP    Customer      PDS           CNXCSTP
CUSTOMRQ    Customer      QSAM PS       CNXCSTQ
CUSTOMRR    Customer      VSAM RRDS      CNXCSTR
CUSTOMRX    Customer      Remote KSDS    CNXCSTX
EQUIPMNE    Equipment    VSAM ESDS      CNXEQE
EQUIPMNP    Equipment    PDS           CNXEQP
EQUIPMNQ    Equipment    QSAM PS       CNXEQQ
EQUIPMNR    Equipment    VSAM RRDS      CNXEQR
EQUIPMNT    Equipment    VSAM KSDS      CNXEKQ
ORDER       Order        VSAM KSDS      CNXORK
ORDERE      Order        VSAM ESDS      CNXORE
ORDERP      Order        PDS           CNXORP
ORDERQ      Order        QSAM PS       CNXORQ
ORDERR      Order        VSAM RRDS      CNXORR

MA a 04/015

```

13. After importing one or more CONNX sample VSAM files, save the information in a CONNX CDD file, define an ODBC data source which points to the new CDD file, and use the [CONNX InfoNaut Query Tool](#) to execute queries against the tables defined in the ODBC data source.

For more information on creating CONNX ODBC data sources and connecting to them with InfoNaut, refer to [Testing a Database Connection](#).

To import metadata for the CONNX sample QSAM customer physical sequential file

1. Click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX Data Dictionary**. The CONNX Data Dictionary Manager appears.
2. Click the **Import** button. The **Import CDD** dialog box appears.

Import CDD

The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX.

OK Cancel

Import Type: VSAM COBOL FD files

Enter a COBOL FD data set member: Browse

Logon Information

Server:

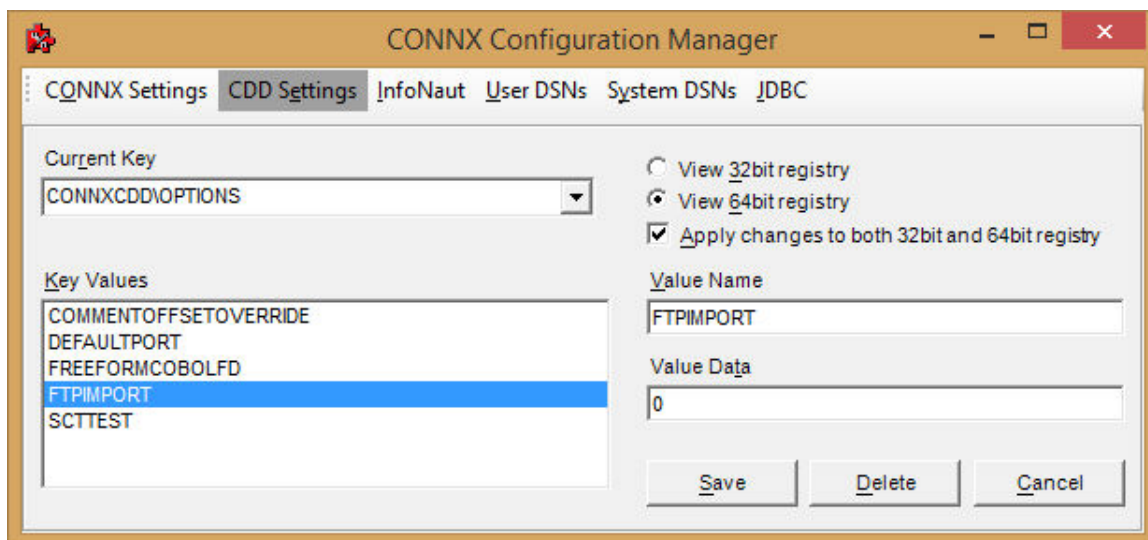
UserName:

Password:

TCP/IP Port:

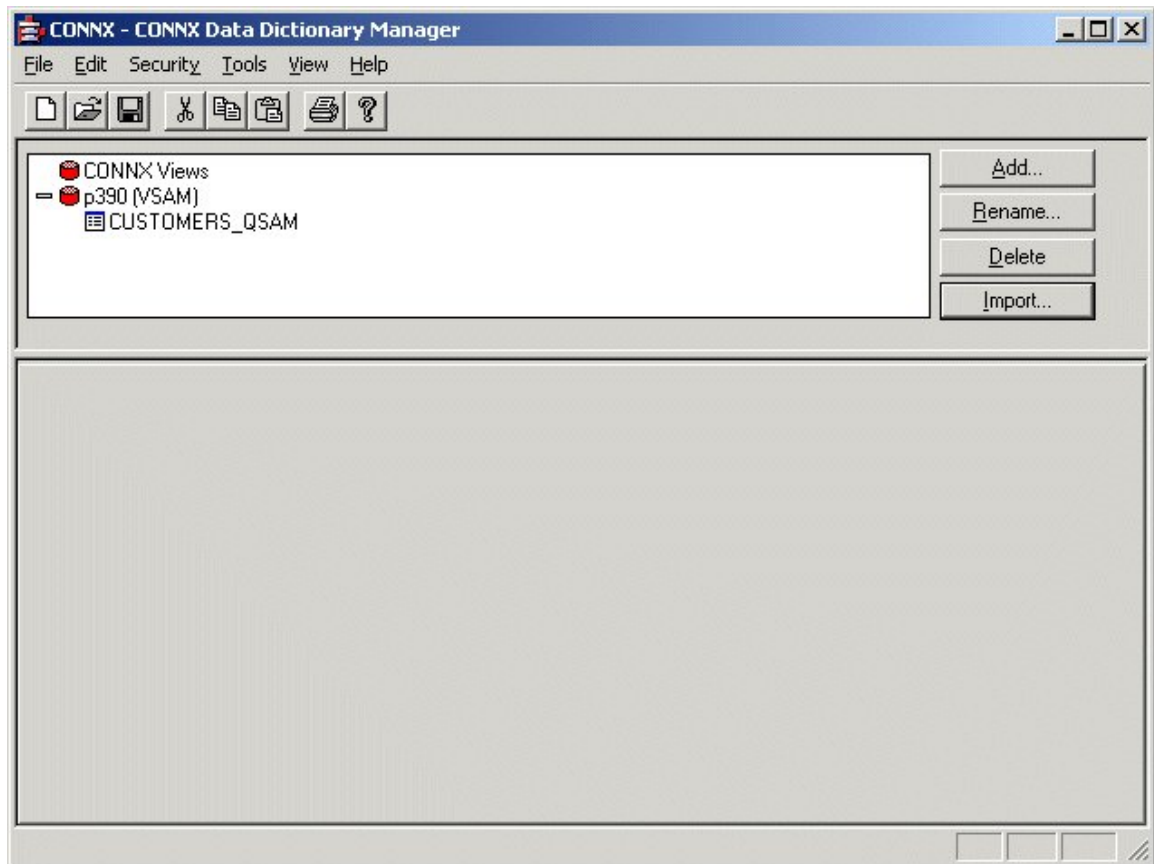
Destination Database: <New Database Container>

3. Select **VSAM COBOL FD files** from the **Import Type** list box.
4. In the **Enter a COBOL FD data set member** text box, enter **CONNX.STASK.COPYBOOK (CUSTOMRQ)**.
By default CONNX will use FTP to import the copybook. To import without FTP add the following configuration setting:

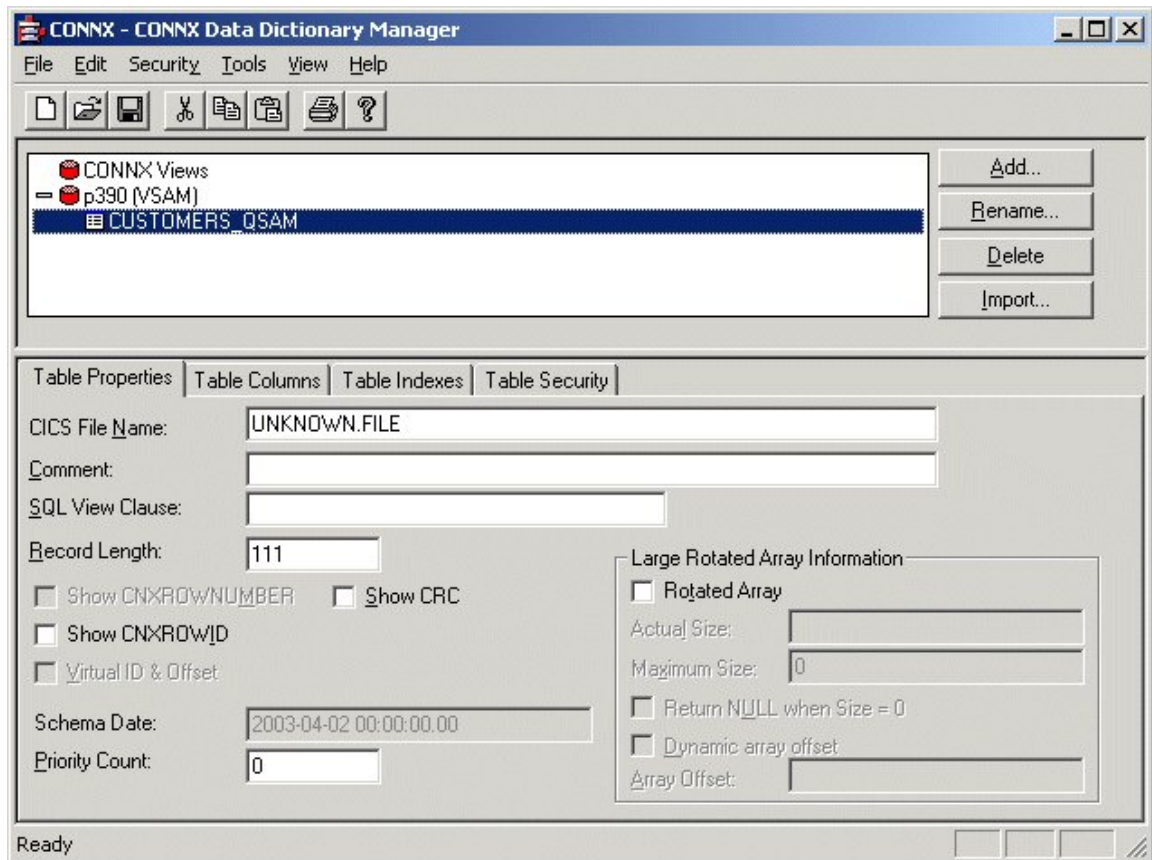


With FTPIMPORT set to zero, the copybook must reside on the PC where the import is being performed (in ascii). Specify the path to the copybook.
Wildcard imports are not permitted in this mode.

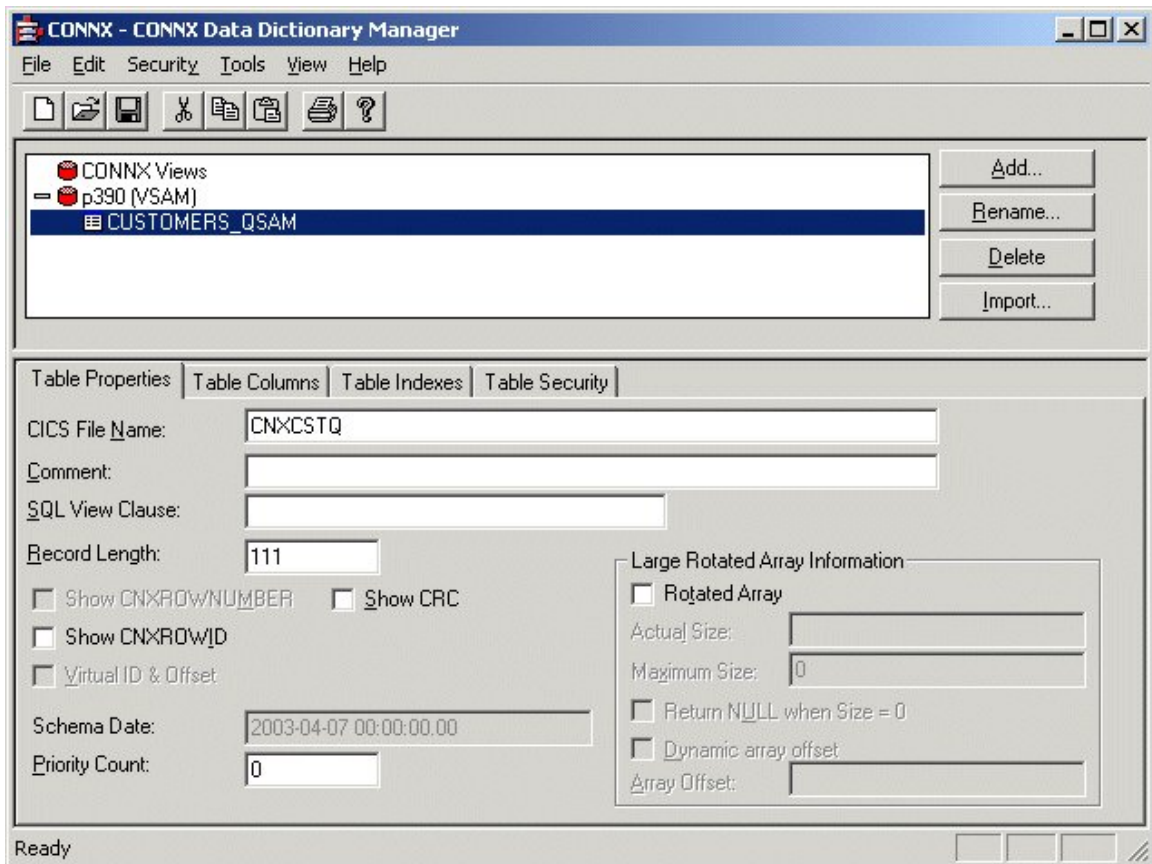
5. In the **Server** text box, enter the symbolic or dotted numeric TCP/IP address of the target host, and then enter your TSO user name and password. Enter a TCP/IP port number.
6. Select a database container in the destination database list box. For more information, see "Add a Database Connection" in the CONNX User Reference Guide, available online, on your CONNX CD-ROM, or within the CONNX product.
7. Click the **OK** button.
8. The CUSTOMERS_QSAM table name and field attributes are imported into the CONNX Data Dictionary Manager. To display the Table Properties tab, click the **CUSTOMERS_QSAM** table icon.



9. The Table Properties tab appears, with the **File Name** text box set to **UNKNOWN.FILE**.

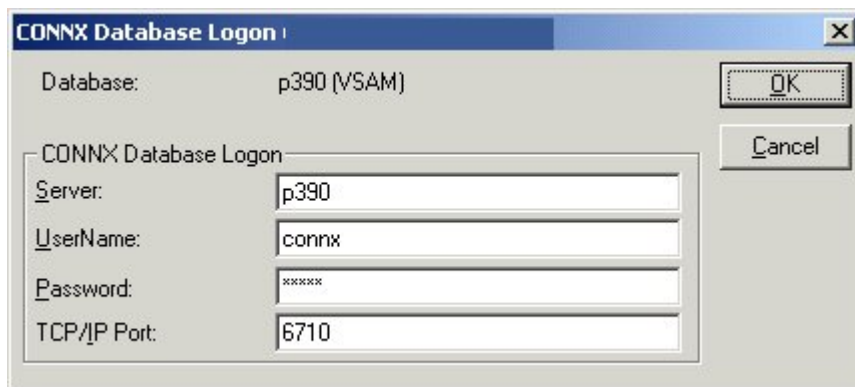


10. For QSAM files, the file layouts are maintained separately from the physical or logical file name (fully-qualified JCL DSN or DDNAME, respectively). The CNXVSAM JCL procedure defines standard DDNAMES for the CONNX sample QSAM files. For the CONNX sample QSAM physical sequential CUSTOMER file, the DDNAME is CNXCSTQ. Overtyping UNKNOWN.FILE with CNXCSTQ:



Note: CNXCSTQ is the default JCL DDNAME for the CONNX sample Customers QSAM physical sequential file.

11. Press <Tab> to open the CONNX Logon dialog box.



12. To complete the import, enter your TSO userid and password, and the correct TCP/IP port number, and click the **OK** button.

13. To import additional CONNX sample QSAM physical sequential files, refer to member @INDEX of CONNX.STASK.COPYBOOK for a cross-reference of VSAM / QSAM / PDS DDNames, and repeat steps 5 through 10.


```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

BROWSE CONNX.STASK.COPYBOOK(@INDEX) - 01.0 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR

***** Top of Data *****
MEMBER      CONNX SAMPLE      CICS FILE NAME /
FILE DESCRIPTION      JCL DDNAME

CUSTOMER    Customer        VSAM KSDS      CNXCSTK
CUSTOMRE    Customer        VSAM ESDS      CNXCSTE
CUSTOMRP    Customer        PDS           CNXCSTP
CUSTOMRQ    Customer        QSAM PS       CNXCSTQ
CUSTOMRR    Customer        VSAM RADS     CNXCSTR
CUSTOMRX    Customer        Remote KSDS   CNXCSTX
EQUIPMNE    Equipment      VSAM ESDS     CNXEQE
EQUIPMNP    Equipment      PDS           CNXEQP
EQUIPMNQ    Equipment      QSAM PS       CNXEQQ
EQUIPMNR    Equipment      VSAM RADS     CNXEQR
EQUIPMNT    Equipment      VSAM KSDS     CNXEKQ
ORDER       Order          VSAM KSDS     CNXORK
ORDERE      Order          VSAM ESDS     CNXORE
ORDERP      Order          PDS           CNXORP
ORDERQ      Order          QSAM PS       CNXORQ
ORDERR      Order          VSAM RADS     CNXORR

MA a 04/015

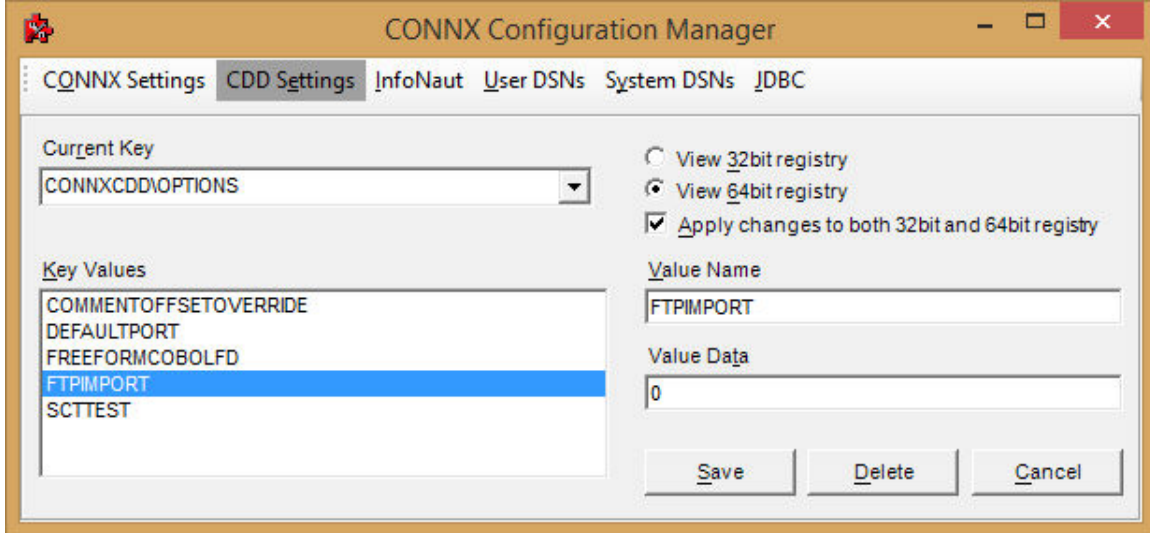
```

14. After importing one or more CONNX sample QSAM physical sequential or partitioned data set files, you can save the information in a CONNX CDD file, define an ODBC data source which points to the new CDD file, and use the [CONNX InfoNaut Query Tool](#) to execute queries against the tables defined in the ODBC data source.

For more information on creating CONNX ODBC data sources and connecting to them with InfoNaut, refer to [Testing a Database Connection](#).

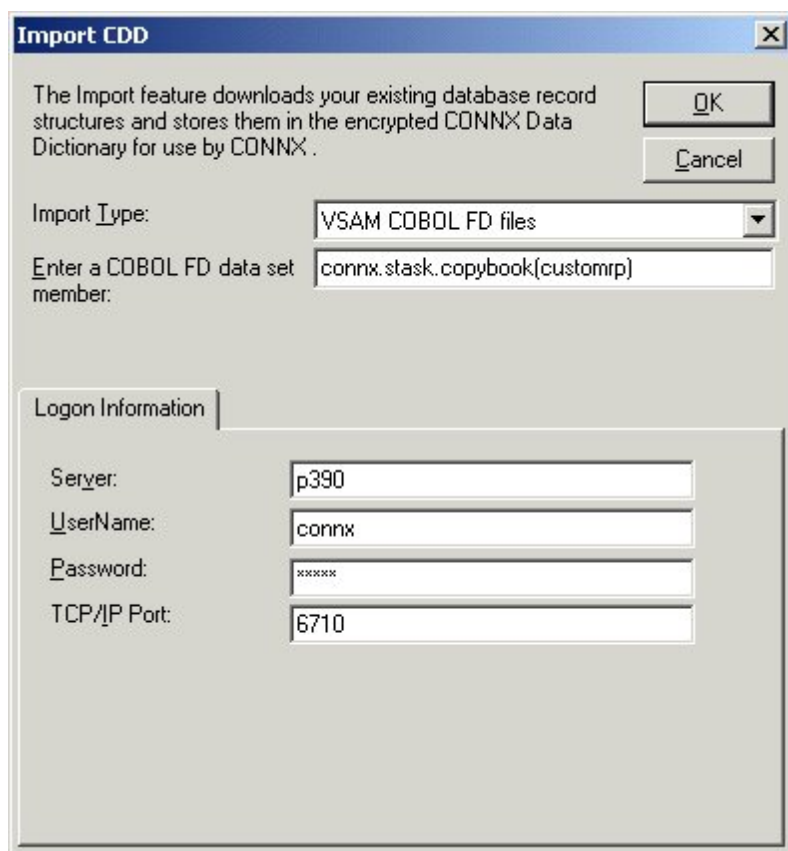
To import metadata for the CONNX sample customer partitioned data set

1. Click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX Data Dictionary**. The CONNX Data Dictionary Manager appears.
2. Click the **Import** button. The **Import CDD** dialog box appears.
3. Select **VSAM COBOL FD files** from the **Import Type** list box.
4. In the **Enter a COBOL FD data set member** text box, enter **CONNX.STASK.COPYBOOK.(CUSTOMRP)**.
By default CONNX will use FTP to import the copybook. To import without FTP add the following configuration setting:



With FTPIMPORT set to zero, the copybook must reside on the PC where the import is being performed (in ascii). Specify the path to the copybook.
Wildcard imports are not permitted in this mode.

5. In the **Server** text box, enter the symbolic or dotted numeric TCP/IP address of the target host, and then enter your TSO user name and password. Enter a TCP/IP port number.
6. Select a database container in the destination database list box. For more information, see Adding a Database Connection in the *CONNX User Reference Guide*, available online, on the CONNX CD-ROM, and within the CONNX product.
7. Click the **OK** button.



Import CDD

The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX.

OK Cancel

Import Type: VSAM COBOL FD files

Enter a COBOL FD data set member: connx.stask.copybook(customrp)

Logon Information

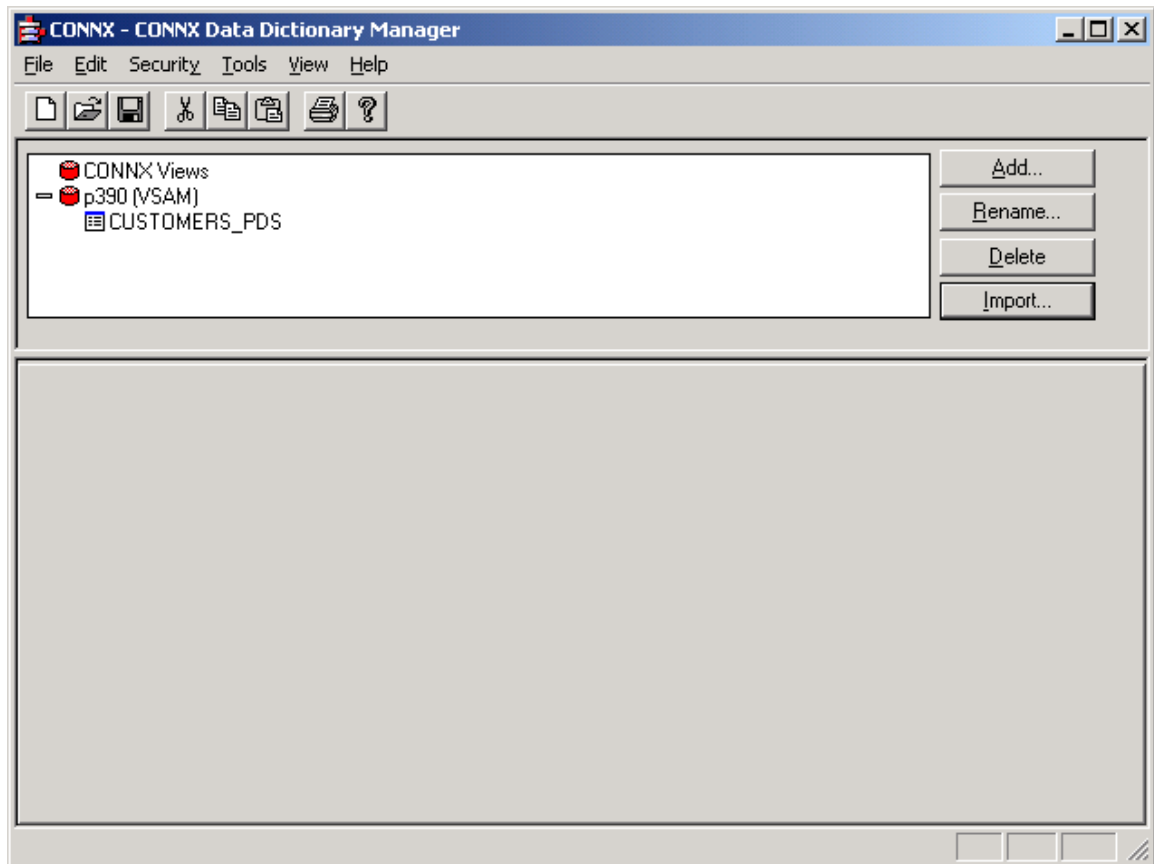
Server: p390

UserName: connx

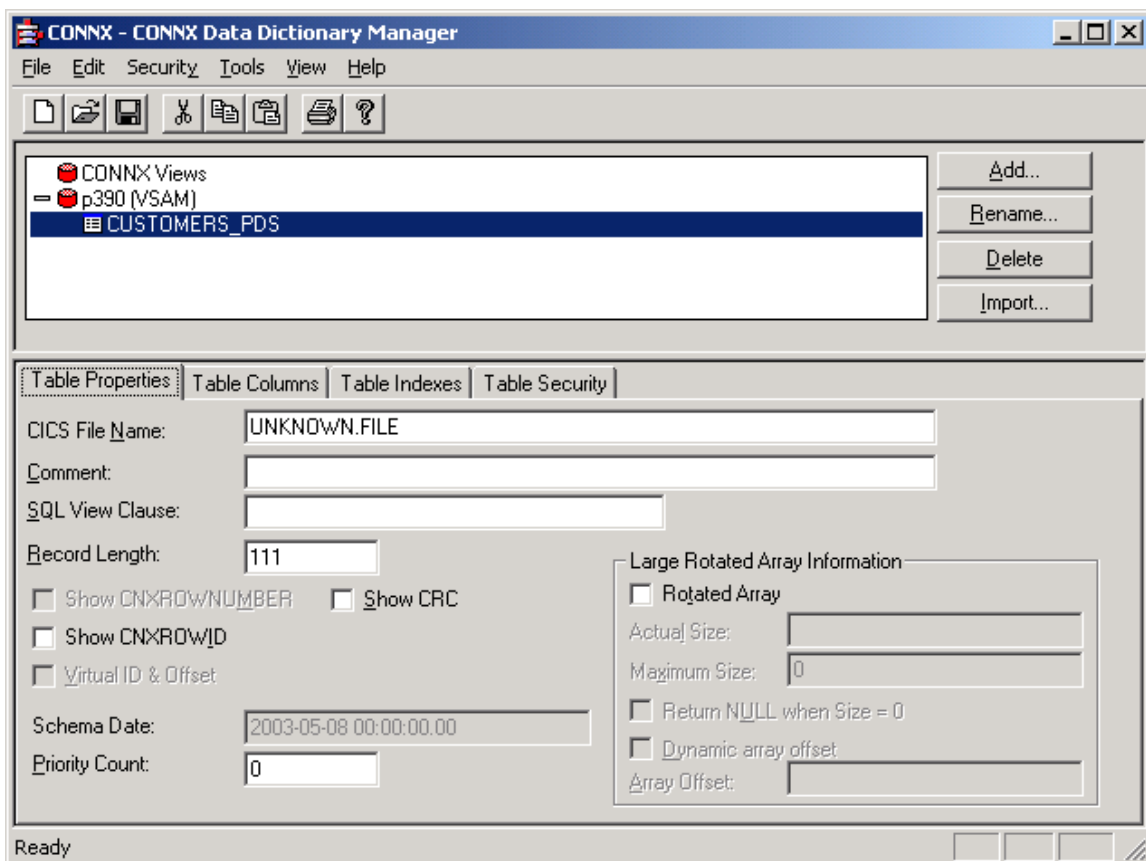
Password: xxxxxx

TCP/IP Port: 6710

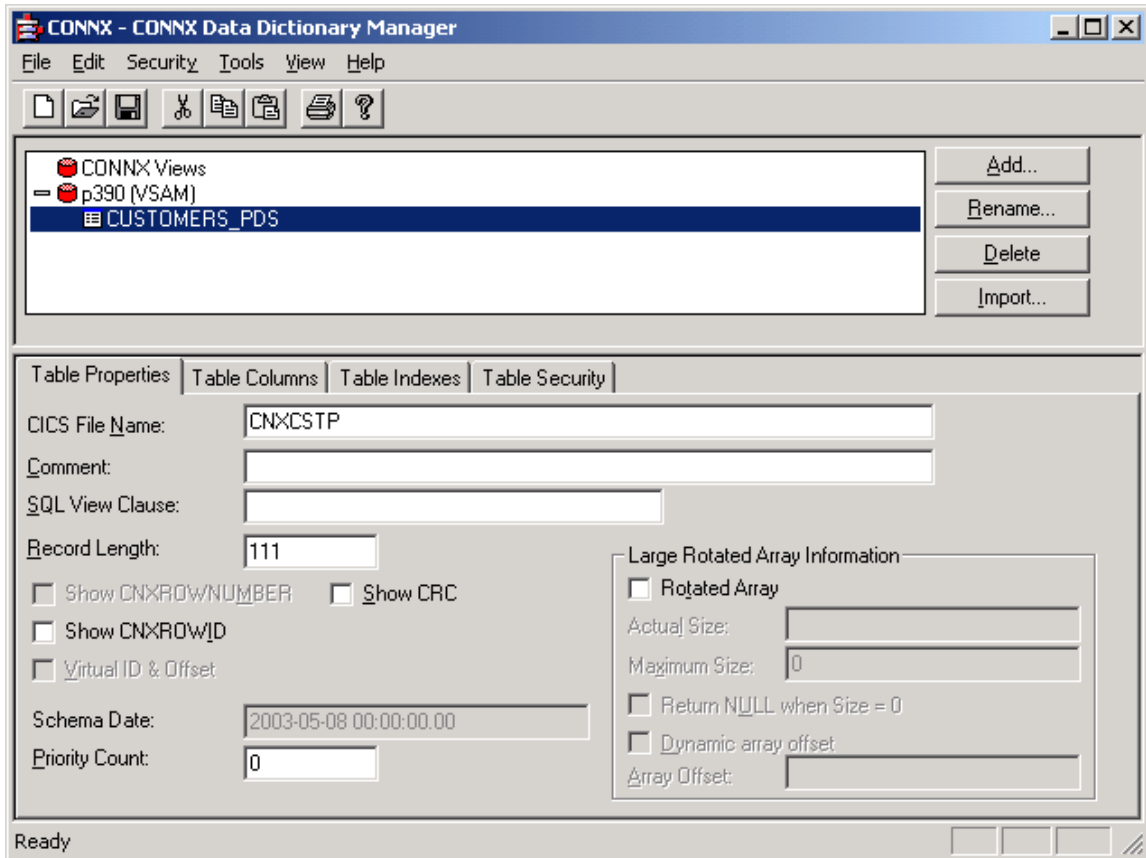
8. The CUSTOMERS_PDS table name and field attributes are imported into the CONNX Data Dictionary Manager. Select the table icon.



9. The **Table Properties** tab appears with the File Name text box set to UNKNOWN.FILE.



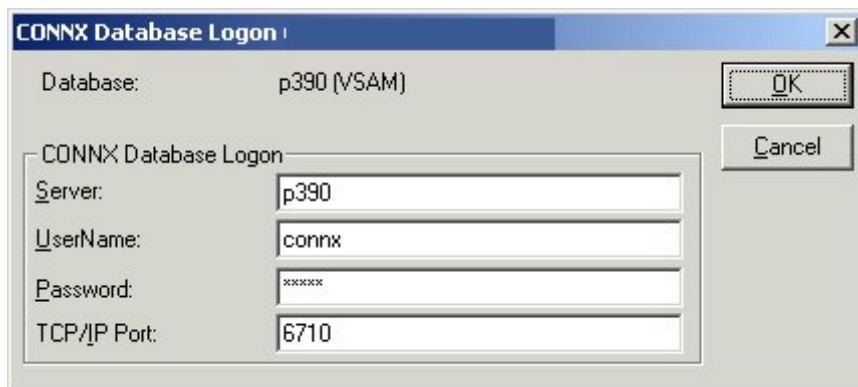
10. For partitioned data set files, the file layouts are maintained separately from the physical or logical file name (fully-qualified JCL DSN or DDNAME, respectively). The CNXVSAM JCL procedure defines standard DDNAMES for the CONNX sample PDS files. For the CONNX sample partitioned data set CUSTOMER file, the DDNAME is CNXCSTP. Overtyping UNKNOWN.FILE with CNXCSTP:



Note: CNXCSTP is the default JCL DDNAME allocated to the CONNX sample Customers QSAM partitioned data set (member) name:

CONNX.STASK.CUSTOMERS.PDS (CUSTOMER)

11. Press <Tab> to open the CONNX Database Logon dialog box.



12. To complete the import, enter your TSO user ID and password, and the correct TCP/IP port number, and then click the **OK** button.
13. To import additional sample CONNX Sample QSAM partitioned data set metadata, refer to CONNX.STASK.COPYBOOK(@INDEX) for a cross-reference of the CONNX sample COBOL copybooks and VSAM / QSAM / PDS JCL DDNAMES, and repeat steps 5 through 10:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Help

BROWSE CONNX.STASK.COPYBOOK(@INDEX) - 01.0 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR

***** Top of Data *****
MEMBER      CONNX SAMPLE      CICS FILE NAME /
FILE DESCRIPTION      JCL DDNAME

CUSTOMER    Customer      VSAM KSDS      CNXCSTK
CUSTOMRE    Customer      VSAM ESDS      CNXCSTE
CUSTOMRP    Customer      PDS            CNXCSTP
CUSTOMRQ    Customer      QSAM PS       CNXCSTQ
CUSTOMRR    Customer      VSAM RADS      CNXCSTR
CUSTOMRX    Customer      Remote KSDS    CNXCSTX
EQUIPMNE    Equipment    VSAM ESDS      CNXEQE
EQUIPMNP    Equipment    PDS            CNXEQP
EQUIPMNQ    Equipment    QSAM PS       CNXEQQ
EQUIPMNR    Equipment    VSAM RADS      CNXEQR
EQUIPMNT    Equipment    VSAM KSDS      CNXEKQ
ORDER       Order        VSAM KSDS      CNXORK
ORDERE      Order        VSAM ESDS      CNXORE
ORDERP      Order        PDS            CNXORP
ORDERQ      Order        QSAM PS       CNXORQ
ORDERR      Order        VSAM RADS      CNXORR

MA a 04/015

```

14. After importing one or more CONNX sample QSAM physical sequential or partitioned data set files, you can save the information in a CONNX CDD file, define an ODBC data source which points to the new CDD file, and use the CONNX InfoNaut Query Tool to execute queries against the tables defined in the ODBC data source.

For more information on creating CONNX ODBC data sources and connecting to them with InfoNaut, refer to [Testing a Database Connection](#).

To stop the CONNX Started Task/Batch Job TCP/IP Listener

In the current example, the CNXVSD0 batch job TCP/IP Listener can be stopped by executing the CNXCFG REXX command, using any of the following syntax options:

1. Implicit syntax from any ISPF command line:

```
tso cnxcfg p(stop 6710)
```

2. Implicit syntax from ISPF option 6:

```
cnxcfg p(stop 6710)
```

Verbose:

```
Exec 'CONNX.STASK.CNTL(CNXCFG)' 'p(stop 6710)'
```

Verbose from any ISPF command line:

```
TSO exec 'CONNX.STASK.CNTL(CNXCFG)' 'p(stop 6710)'
```

2. A successful stop attempt displays the following messages on the invoking terminal screen:

```
Stopping CONNX Started Task VSAM TCP/IP Listener on port 6710.
```

```
Wed Apr 2 14:30:05 2003
```

```
Using TCPIP.
```

```
Wed Apr 2 14:30:05 2003
```

```
Connecting to port (6710) (127.0.0.1)
```

```
CONNX Started Task VSAM TCP/IP Listener Stopped
```

To test CONNX for VSAM query/import functionality against SCT VSAM master files

1. Browse CONNX.STASK.CNTL and find members CNXVSCT, CNXVSCT1, CNXVSCTJ, and CNXVSJ02. Members CNXVSCT and CNXVSCT1 define JCL procedures which invoke the CONNX Started Task TCP/IP Listener/Server programs. Procedure CNXVSCT includes parameters and DDNAMEs for the SCT Master VSAM and CONNX sample VSAM / QSAM / PDS files. Procedure CNXVSCT1 defines parameters and DDNAMEs for the SCT Master VSAM files and omits the DDNAMEs for the CONNX sample files. The batch job execution JCL for these procedures is in members CNXVSCTJ and CNXVSJ02, respectively. See Steps 5 and 6 in [To verify installation of CONNX Started Task components](#) for the recommended settings for the CONNX sample JCL procedure parameters and environment variables.
2. Change the CONNX environment variables in member CNXPARMS from DEBUG=0 to DEBUG=1 and from CNXNOPREAUTHORIZE=0 to CNXNOPREAUTHORIZE=1.
3. Review the following SCT-specific procedure parameters:

ADSHLQ	SCT Alumni Development System VSAM master file high-level qualifier Default:SCT.ADSBASE
FRSHLQ	SCT Financial Records System VSAM master file high-level qualifier Default: SCT.FRSBASE
HRSHLQ	SCT Human Resources System VSAM master file high-level qualifier Default: SCT.HRSBASE
LMSHLQ	SCT Loan Management System VSAM master file High-Level Qualifier Default: SCT.LMSBASE
SISHLQ	SCT Student Information Systems VSAM master file High-Level Qualifier Default: SCT.SISBASE
ZSSHLQ	SCT Z Support Software VSAM master file High-Level Qualifier Default: SCT.ZSSBASE

Note: For testing purposes, it is not necessary to define all SCT subsystem VSAM master files in the CNXVSCT/CNSVSCT1 procedure. You can make a backup copy of CNXVSCT or CNXVSCT1 and remove or comment out the SCT symbolic parameters and DDNAMEs which are not installed at your site.

4. Edit member CNXVSCTJ or CNXVSJ02, change the job card parameters to match your installation standards, and submit the job to a batch execution queue. The CONNX TCP/IP Listener program (CNXRUNB) starts execution and attempts to listen on the TCP/IP port defined by the CNXRUNPORT environment variable. The following CONNX debug tracing messages, which are by default written to the CNXVSCT job output queue, show a successful attempt to listen on port 6710:

```
Tue Apr  2 17:59:21 2003
  Listening to network port 6710.
Tue Apr  2 17:59:26 2003
  Binding to port 6710
Tue Apr  2 17:59:26 2003
  NET TCP Listening on (0.0.0.0) Port(6710)
```


5. After these messages are written to the job log, the CNXVSCT job waits for a connect request from an ODBC / JDBC / OLE DB / .NET application. An SDSF display of an idle CNXVSCT batch server job follows:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 0 SIO 3 CPU 12 LINE 1-1 (1)
COMMAND INPUT ==>
PREFIX=CNXVSCT DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAM PROCSTEP JOBID OWNER C POS DP PGN REAL PAGING S
CNXVSCT CNXVSCT CNXRUNB JOB07088 IBMUSER A LD FF 1 493 0.00 0.

MA a 05/021

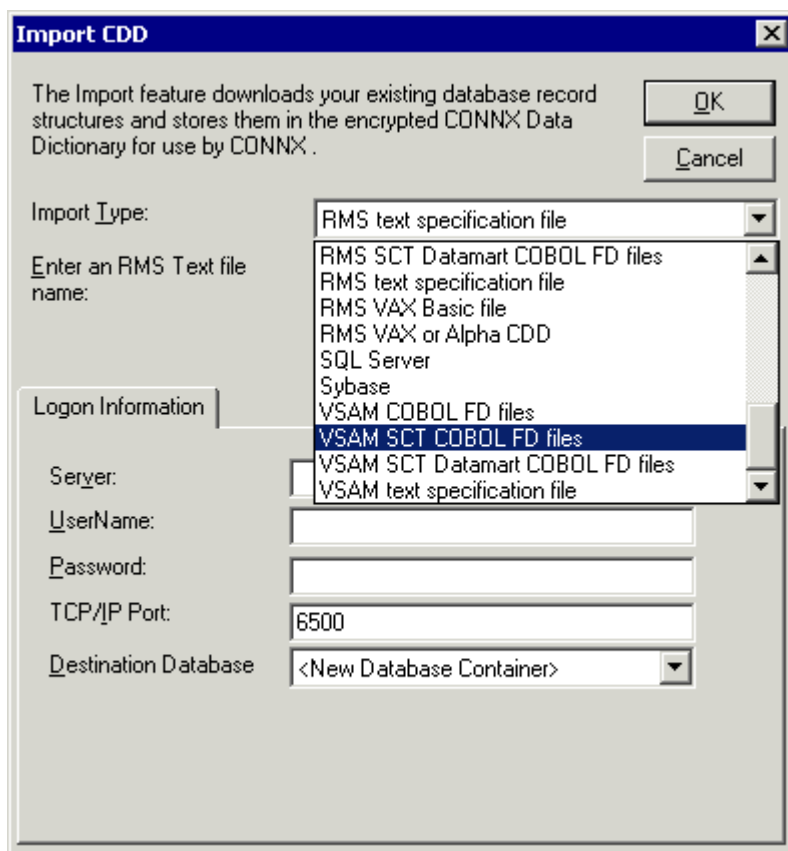
```

6. Refer to the table above for the expected tracing messages for a successful bind to the designated TCP/IP port (default: CNXRUNPORT=6500 in member CNXPARMs of CONNX.STASK.CNTL).

To test CONNX for VSAM Started Task / Batch Job SCT metadata import functionality

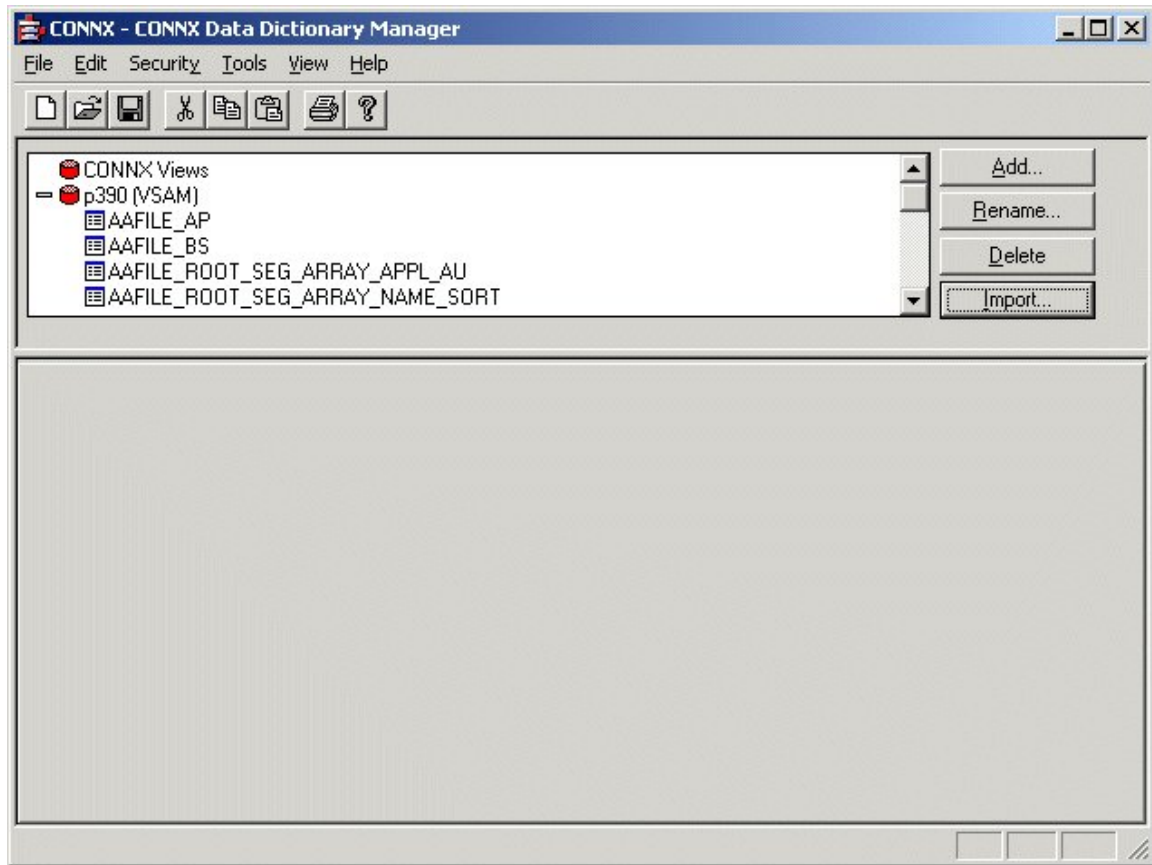
You can use the CONNX Administrator to import SCT metadata from the SCT COBOL master file copybooks. The following shows how to import metadata for the SCT VSAM master AAFILE.

1. Click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX Data Dictionary**. The CONNX Data Dictionary Manager appears.
2. Click the **Import** button. The Import CDD dialog box appears.
3. To process SQL queries against SCT VSAM master files, the metadata defined in SCT COBOL copybooks must be imported into a CONNX CDD. Select **VSAM SCT COBOL FD files** from the Import Type list box.

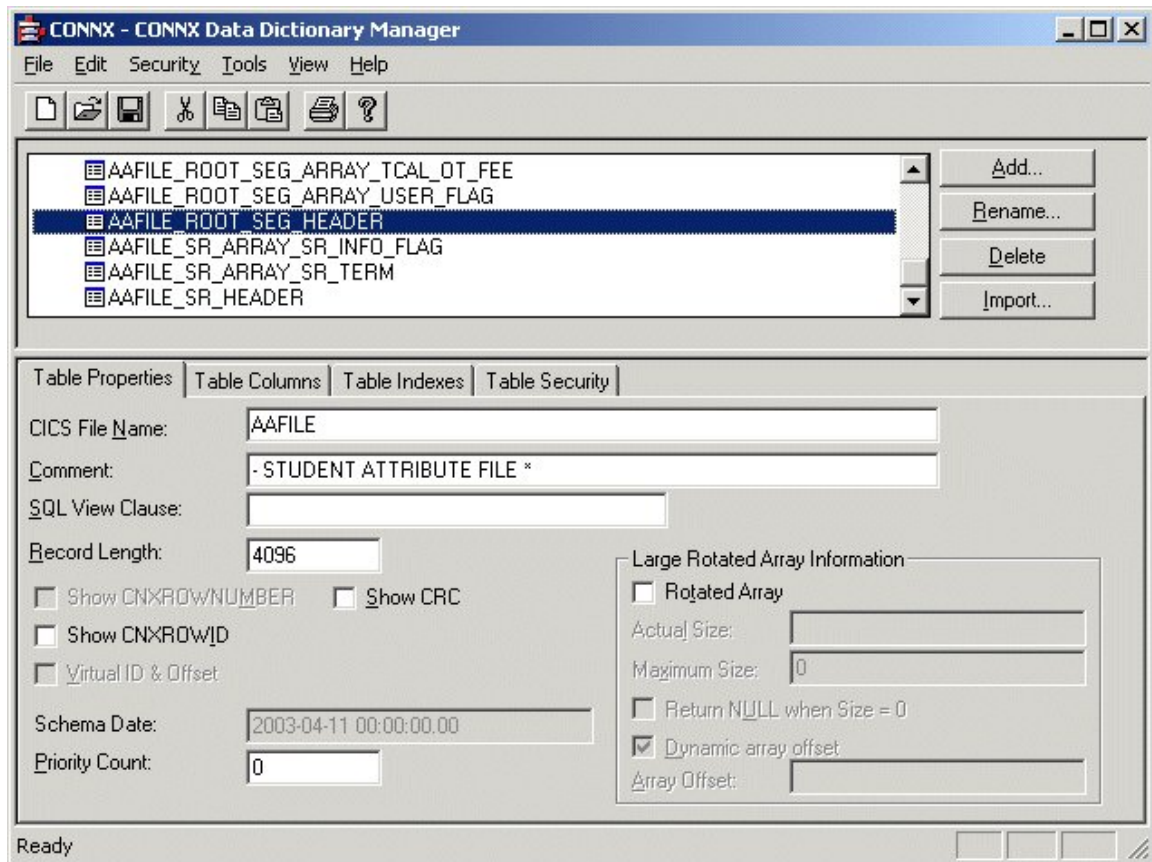


4. In the **Enter an SCT COBOL FD data set member** text box, enter a fully qualified partitioned data set (member) name; for example, the default file layout for the SCT AAFILE resides in SCT.SISBASE.SOURCE(ACAARC).
5. In the **Server** text box, enter the dotted numeric or symbolic TCP/IP address for the target system, and then enter your TSO user name and password.
6. Enter a **TCP/IP** port number.
7. Select a database container in the destination database list box. For more information, see Adding a Database Connection in the *CONNX User Reference Guide*, available online, on your CONNX CD-ROM, and within the CONNX product, and then click the **OK** button.
8. The CONNX Data Dictionary Manager imports the AAFILE record layouts, automatically detects COBOL REDEFINES clauses, and creates one or more virtual tables for each record segment. For SCT COBOL copybooks, the CONNX Data Dictionary Manager automatically detects the JCL

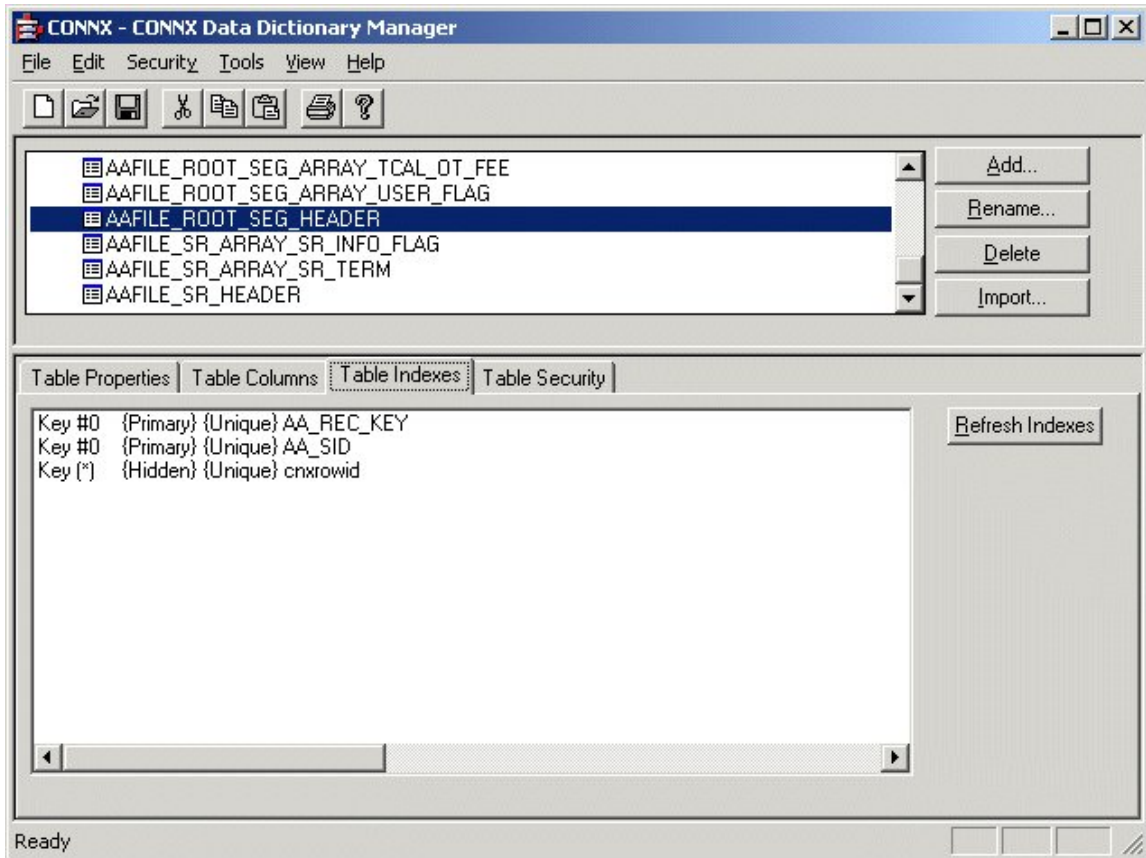
DDNAME (AAFILE), and imports file attributes such as key offset and length and index field name(s) from the corresponding JCL physical data set name.



9. Select a table. The **Table Properties** tab appears.



10. To display the primary and alternate keys, click the **Table Indexes** tab.



Note: For SCT Master file copybooks, the CONNX CDD Administrator automatically detects the DDNAMEs of the corresponding SCT VSAM Master files and sets the File Name text box to the DDNAME. After importing one or more SCT VSAM master files, you can save the information in a CONNX CDD file, define an ODBC data source which points to the new CDD file, and use the CONNX InfoNaut application to execute queries against the new data source files.

11. Refer to [To stop the CONNX Started Task/Batch Job TCP/IP Listener](#) for instructions on how to stop the listener process.

Note: This scenario assumes that the started tasks/batch jobs CNXVSAM and CNXVSCT are not running simultaneously. These tasks/jobs are not mutually exclusive, but running both simultaneously requires two different values for the =CNXRUNPORT environment variable. Running the CNXVSAM and CNXVSCT batch jobs simultaneously requires pointing each JCL procedure to a separate CNXPARMs PDS control record member in CONNX.STASK.CNTL. To do so, copy member CNXPARMs to a new member in the *.CNTL PDS; then change the =CNXRUNPORT=nnnn entry to point to a different TCP/IP port. Next, change the CNXPARMs symbolic JCL parameter in either procedure CNXVSAM or CNXVSCT to point to the new PDS member; for example:

CNXPARMs=CNXPARMZ

To test the CONNX for VSAM Started Task / Batch Job User ID/Password

The previous scenarios test the CONNX for VSAM Started Task/Batch Job Import and Query functionality without userid/password verification. Enabling the userid / password verification logic requires that the CONNX.STASK.LOAD library be APF-authorized.

1. If you have systems programmer authority, you can enter the following operator command from the system console or from the SDSF command line:

```
SETPROG APF,ADD,DSNAME=CONNX.STASK.LOAD,VOLUME=volume
```

Where volume is the DASD volume serial on which CONNX.STASK.LOAD resides.

If this command succeeds, the following message is displayed in the system log:

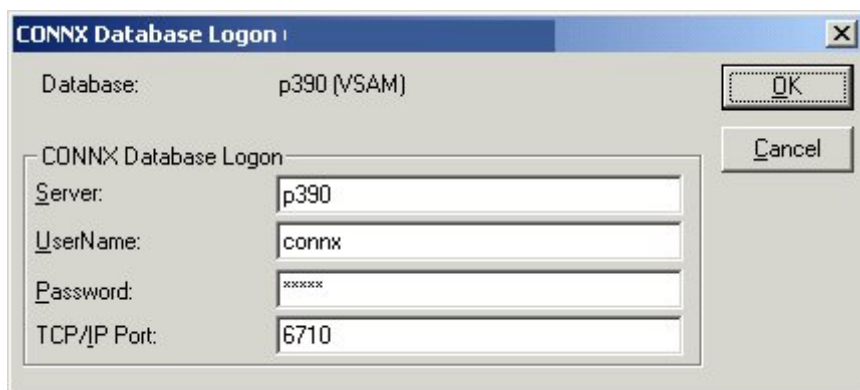
```
CSV410I DATA SET CONNX.STASK.LOAD ON VOLUME volume ADDED TO APF LIST
```

2. Edit member CNXPARMS of CONNX.STASK.CNTL and make the following CONNX environment variable entries:

```
DEBUG=1
```

```
CNXNOPREAUTHORIZE=0
```

3. Edit member CNXVSAMJ, change the job card parameters to match your installation standards, and submit the job to a batch execution queue. Verify that the CONNX Listener Program CNXRUNB successfully binds to the specified CNXRUNPORT value.
4. Use the [InfoNaut query tool](#) to connect to a previously defined ODBC, JDBC, OLE DB, or .NET data source. First, use a valid userid/password combination; next, try a non-existent userid/password. The first attempt should fail and display the **CONNX Database Logon** dialog box.



5. A second attempt with the same userid / password combination displays the following message box:

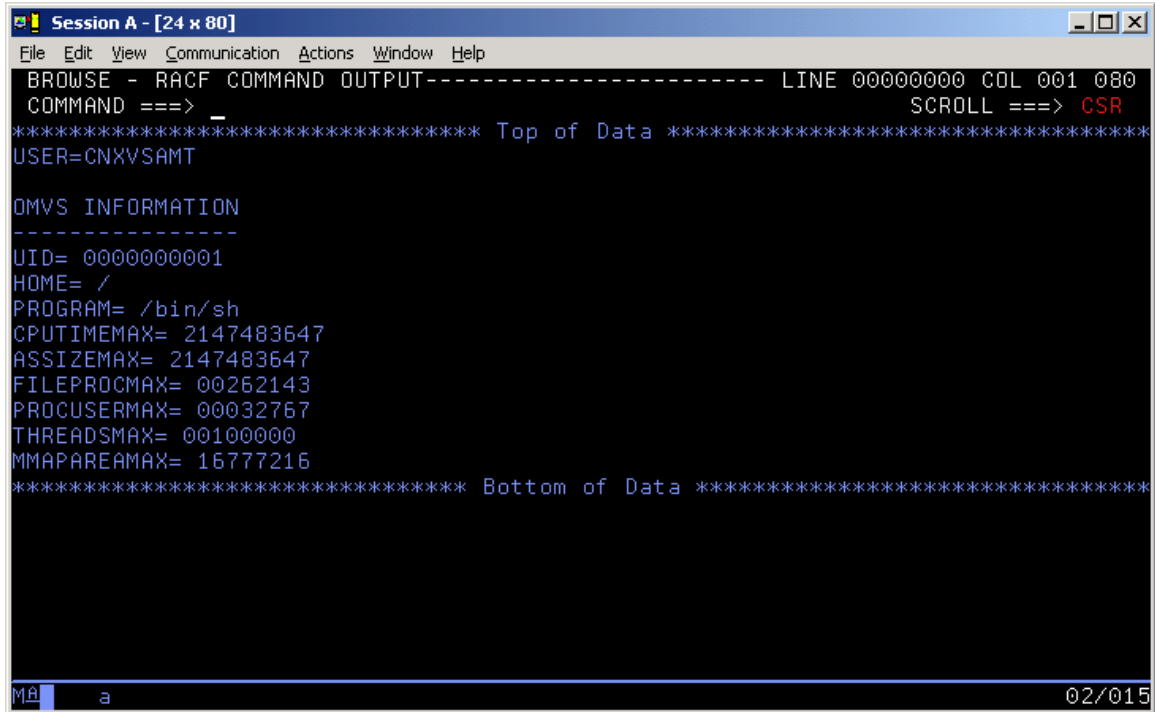


Caution: Test the erroneous userid / password combination with a non-existent userid. Repeated tests with a valid userid and the wrong password can potentially disable your userid.

6. After you have verified that procedures CNXVSAM and CNXVSCTJ work as batch jobs, you can copy and/or rename one or both to a system started procedure PDS, e.g. SYS1.PROCLIB. As a final

installation step, ask your security administrator or systems programmer to create a RACF user ID for each procedure copied to the production started procedure PDS.

For example, the CONNX Development and Testing staff work with two different copies of CNXVSAM (CNXVSAMD and CNXVSAMT) in SYS1.PROCLIB, and RACF user IDs are defined to match the procedure names:



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
BROWSE - RACF COMMAND OUTPUT----- LINE 00000000 COL 001 080
COMMAND ==> _ SCROLL ==> CSR
***** Top of Data *****
USER=CNXVSAMT

OMVS INFORMATION
-----
UID= 0000000001
HOME= /
PROGRAM= /bin/sh
CPUTIMEMAX= 2147483647
ASSIZEMAX= 2147483647
FILEPROCMA= 00262143
PROCUSERMA= 00032767
THREADSMA= 00100000
MMAPAREMA= 16777216
***** Bottom of Data *****

MA a 02/015

```

Note: There are many advantages to creating CONNX started procedures, but this step can be bypassed if the alternative of running CONNX as one or more batch jobs provides adequate response time and security.

Part 3: CONNX TSO Configuration Utility

Introduction to the CONNX TSO Configuration Utility

The CONNX TSO configuration utility for VSAM (REXX exec CNXCFG and program CNXCFGB) enables the user to to set, remove, and display persistent CONNX environment variables and to stop the CONNX TCP/IP Listener Program (CNXRUNB). Unlike its CICS counterpart, the CONNX TSO configuration utility cannot be used to start the CONNX TCP/IP Listener Program (CNXRUNB). This program is started as a job step in the CNXVSAM/CNXVSCT JCL procedures, which can be executed either as batch jobs or as started tasks.

To stop the CONNX Started Task/Batch Job TCP/IP Listener

To terminate the CONNX Started Task VSAM TCP/IP Listener gracefully, use the REXX procedure CNXCFG. This is the primary function of the CONNX TSO Configuration Utility.

Important: Using this utility to stop the CONNX Started Task TCP/IP Listener will cause the CNXRUNB program and its started task or batch job to exit. When CNXRUNB terminates, it halts all instances of the CONNX VSAM Started Task server program (CNXVSD0B), which run as sub-tasks in the same address space. A separate instance (sub-task) of CNXVSD0B is created for each CONNX client connection via ODBC, JDBC, OLE DB or .NET to the specified TCP/IP address + port. Therefore, stopping the CNXRUNB program will sever any active CONNX connections to the CNXVSD0B server sub-tasks.

For example, the CNXVSAM Started Task/Batch Job TCP/IP listener is started and listening on port 6710.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 2 SIO 2 CPU 14 LINE 1-1 (1)
COMMAND INPUT ==> SCROLL ==> CSR
PREFIX=CNXVSAM DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAM PROCSTEP JOBID OWNER C POS DP PGN REAL PAGING S
CNXVSAM CNXVSAM CNXRUNB JOB07073 IBMUSER A LO FF 1 451 0.00 0.
MA a 04/021

```

Three connections are currently active from three different PC clients. The output from a TSO NETSTAT command displays the CONNX TCP/IP Listener connection and the client connections:

```

EZZ2587I CNXVSAM 000026F0 0.0.0.0..6710 0.0.0.0..0
Li
sten
EZZ2587I CNXVSAM 00002709 123.123.123.123..6710 123.123.123.250..2381
Est
ablsh
EZZ2587I CNXVSAM 00002 70B 123.123.123.123..6710 123.123.123.251..2384
Est
ablsh
EZZ2587I CNXVSAM 00002704 123.123.123.123..6710 123.123.123.252..2375
Est
ablsh

```

Executing the REXX CNXCFG command with the parameter string p(stop 6710) will stop all of the above connections, as well as the CNXVSAM Started Task / Batch Job. The implicit and verbose forms of the cnxcfg p(stop) command are:

```
From the ISPF Command Shell:
cnxcfg p(stop 6710)
or
exec 'connx.v08r09.stask.cntl(cnxcfg)' 'p(stop 6710)'
From any ISPF Command line:
tso cnxcfg p(stop 6710)
or
tso exec 'connx.v08r09.stask.cntl(cnxcfg)' 'p(stop 6710)'
```

For a successful stop request, the following messages are displayed on the invoking terminal:

```
Parms = 'STOP 6710'
loadlib = 'CONNX.V08R09.STASK.load'
file = 'CONNX.VXXRXX.STASK.cnxcfg'
Stopping CONNX Started Task VSAM TCP/IP Listener on port 6710.
Thu Apr  3 14:17:55 2003
Using TCPIP.
Thu Apr  3 14:17:55 2003
  Connecting to port (6710) (127.0.0.1)
CONNX Started Task VSAM TCP/IP Listener Stopped  .
```

CONNX TSO Configuration Utility and persistent environment variables

The CONNX for VSAM / QSAM / PDS Started Task product supports two methods for setting environment variables. The first and easier of the two methods, discussed in section 3.4.1, is to enter the environment variables as standard variable / value pairs (=variable =value) in separate records in a partitioned data set member. Refer to the discussion of the pre-set environment variables in member CNXPARMs of CONNX.VVRR.STASK.CNTL. Environment variables set using this approach are passed in the parameter string of the EXEC PGM=CNXRUNB statement in the CNXVSAM / CNXVSCT JCL procedures. For example, starting two instances of the CNXRUNB Listener requires two separate CNXPARM members so that two unique TCP/IP ports can be defined via the =CNXRUNPORT environment variable. Similarly, the CONNX tracing logic can be enabled in one CNXPARMs member and disabled in another via the =DEBUG environment variable.

The second method of managing CONNX environment variables is via the CNXCFG REXX command file, which invokes program CNXCFGB to update the CNXCFG VSAM KSDS. Using this method, you can insert, update and delete persistent CONNX environment variables which reside in a key-sequenced VSAM file (CONNX.VVRR.STASK.CNXCFG). The CONNX environment variables defined via this method can be shared across multiple instances of the CNXVSAM/CNXVSCT started task(s) or batch job(s).

To set a persistent CONNX environment variable

1. The following implicit and verbose syntax can be used to set a persistent CONNX environment variable:

Implicit:

From any ISPF command shell:

```
cnxcfg p(variable-name value)
```

or Verbose:

```
exec 'connx.stask.cntl(cnxcfg)' 'p(variable-name value)'
```

Implicit:

From any ISPF command line:

```
tso cnxcfg p(variable-name value)
```

or Verbose:

```
tso exec 'connx.stask.cntl(cnxcfg)' 'p(variable-name value)'
```

For instance, to set the persistent CONNX environment variable DEBUG to 1, the implicit ISPF command line syntax is:

```

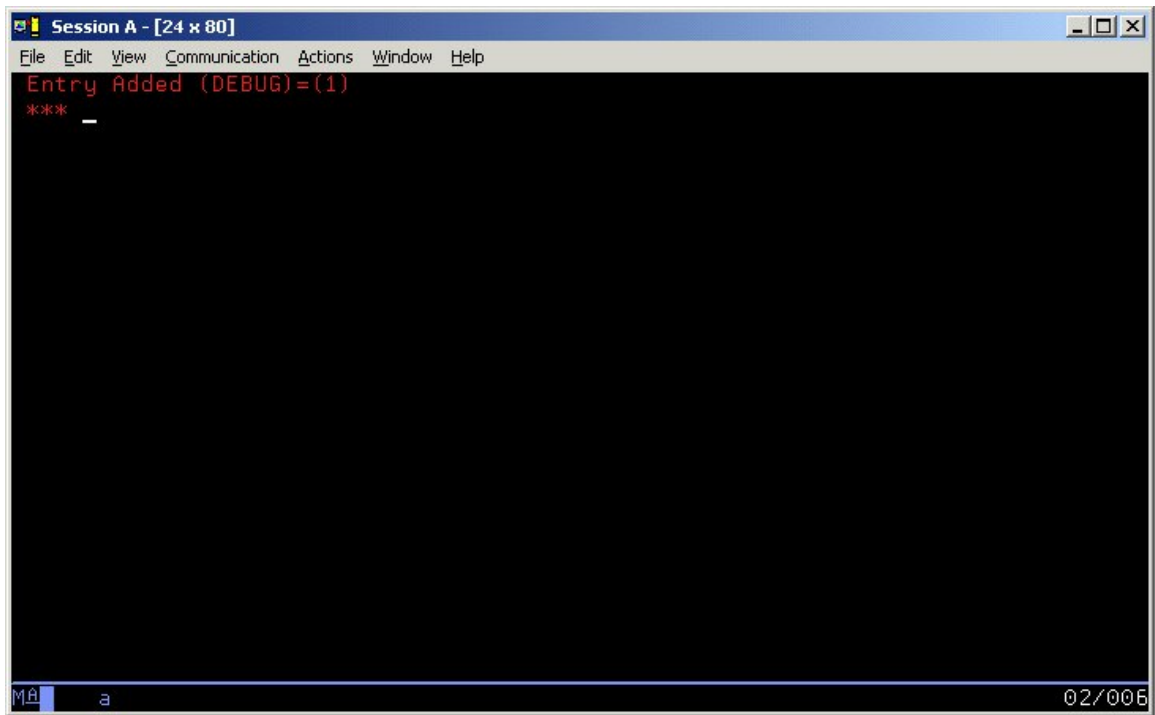
Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu
Option ==> tso cnxcfg p(debug 1)

0 Settings      Terminal and user parameters      User ID . : IBMUSER
1 View          Display source data or listings    Time. . . : 19:09
2 Edit          Create or change source data      Terminal. : 3278
3 Utilities     Perform utility functions         Screen. . : 1
4 Foreground    Interactive language processing   Language. : ENGLISH
5 Batch         Submit job for language processing  Appl ID . : ISR
6 Command       Enter TSO or Workstation commands   TSO logon : ISPFPROC
7 Dialog Test   Perform dialog testing            TSO prefix:
8 LM Facility   Library administrator functions   System ID : P390
9 IBM Products  IBM program development products   MVS acct. : ACCT#
10 SCLM         SW Configuration Library Manager   Release . : ISPF 4.8
11 Workplace    ISPF Object/Action Workplace
M More         Additional IBM Products

Enter X to Terminate using log/list defaults
Parms = 'DEBUG 1'
loadlib = 'CONNX.STASK.load'
file = 'CONNX.STASK.cnxcfg'
***
MA a 24/005

```



To change a CONNX persistent environment variable

- Using the implicit syntax and the DEBUG variable as an example, enter the following:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

                ISPF Primary Option Menu
Option ==> tso cnxcfg p(debug 0)

0 Settings          Terminal and user parameters          User ID . : IBMUSER
1 View             Display source data or listings       Time. . . : 19:09
2 Edit            Create or change source data         Terminal. : 3278
3 Utilities        Perform utility functions           Screen. . : 1
4 Foreground      Interactive language processing       Language. : ENGLISH
5 Batch           Submit job for language processing    Appl ID . : ISR
6 Command         Enter TSO or Workstation commands     TSO logon : ISPFPROC
7 Dialog Test     Perform dialog testing                TSO prefix:
8 LM Facility     Library administrator functions          System ID : P390
9 IBM Products   IBM program development products        MVS acct. : ACCT#
10 SCLM          SW Configuration Library Manager      Release . : ISPF 4.8
11 Workplace     ISPF Object/Action Workplace
M More          Additional IBM Products

Enter X to Terminate using log/list defaults
Parms = 'DEBUG 0'
loadlib = 'CONNX.STASK.load'
file = 'CONNX.STASK.cnxcfg'
***
MA a 24/006

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

Removing Configuration Entry (DEBUG)
Configuration Entry Successfully Removed. (DEBUG)
Entry Added (DEBUG)=(0)
***
MA a 04/006

```

To change the number of the CONNX TCP/IP port persistent environment variable

1. The default value of the TCP/IP port on which the CONNX Started Task VSAM TCP/IP program listens is 6500. To change this persistent environment variable, the following implicit syntax can be used:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

                ISPF Primary Option Menu
Option ==> tso cnxcfg p(cnxrunport 6710)

0 Settings      Terminal and user parameters      User ID . : IBMUSER
1 View         Display source data or listings   Time. . . : 19:09
2 Edit        Create or change source data     Terminal. : 3278
3 Utilities    Perform utility functions       Screen. . : 1
4 Foreground  Interactive language processing  Language. : ENGLISH
5 Batch       Submit job for language processing Appl ID . : ISR
6 Command     Enter TSO or Workstation commands TSO logon : ISPFPROC
7 Dialog Test Perform dialog testing          TSO prefix:
8 LM Facility Library administrator functions  System ID : P390
9 IBM Products IBM program development products MVS acct. : ACCT#
10 SCLM       SW Configuration Library Manager Release . : ISPF 4.8
11 Workplace  ISPF Object/Action Workplace
M More       Additional IBM Products

Enter X to Terminate using log/list defaults
Parms = 'CNXRUNPORT 6710'
loadlib = 'CONNX.STASK.load'
file = 'CONNX.STASK.cnxcfg'
***
MA a 24/005

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

Removing Configuration Entry (CNXRUNPORT)
Configuration Entry Successfully Removed. (CNXRUNPORT)
Entry Added (CNXRUNPORT)=(6710)
***
MA a 04/005

```

2. If the port number is changed, all clients which connect to the Started Task/Batch Job which listens on this port must be updated to the same port number. See TCP/IP Connection Port in the *CONNX User Reference Guide*, available online, on the CONNX CD-ROM, and within the CONNX product, for more information on changing the TCP/IP connection port for client machines.

To convert the persistent CONNX port number to the default

- Using the implicit execution syntax, enter **tso cnxcfg p(cnxrunport)**

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu
Option ==> tso cnxcfg p(cnxrunport)

0 Settings      Terminal and user parameters      User ID . : IBMUSER
1 View         Display source data or listings   Time. . . : 19:09
2 Edit        Create or change source data   Terminal. : 3278
3 Utilities    Perform utility functions         Screen. . : 1
4 Foreground   Interactive language processing    Language. : ENGLISH
5 Batch        Submit job for language processing  Appl ID . : ISR
6 Command      Enter TSO or Workstation commands     TSO logon : ISPFPROC
7 Dialog Test  Perform dialog testing              TSO prefix:
8 LM Facility  Library administrator functions     System ID : P390
9 IBM Products IBM program development products  MVS acct. : ACCT#
10 SCLM        SW Configuration Library Manager   Release . : ISPF 4.8
11 Workplace   ISPF Object/Action Workplace
M More        Additional IBM Products

Enter X to Terminate using log/list defaults
Parms = 'CNXRUNPORT'
loadlib = 'CONNX.STASK.load'
file = 'CONNX.STASK.cnxcfg'
***
MA a 24/006

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

Removing Configuration Entry (CNXRUNPORT )
Configuration Entry Successfully Removed. (CNXRUNPORT )
***
MA a 03/006

```

CONNX Environment Variables

CONNX Environment Variables: Scope and Precedence

The CONNX environment variable search logic first searches for variables defined in the CNXPARMS PDS member, and then queries the environment variables stored in the CNXCFG VSAM KSDS file. To define persistent environment variables across multiple instances of the CNXVSAM/CNXVSCT started task(s) or batch job(s), use the CNXCFG REXX exec file.

To define unique CONNX environment variables per Started Task/Batch Job

1. To define CONNX environment variables specific to a single Started Task/Batch Job using a PDS member such as CNXPARMs, create a new member in the CONNX.STASK.CNTL PDS, and enter the environment variables as =variable=value. Next, change the Started Task/Batch Job JCL symbolic parameter CNXPARMs to the new member name; for example:

```
//CNXPARMs=CNXPARMZ
```

which substitutes as:

```
//CNXPARMs DD DSN=CONNx.STASK.CNTL(CNXPARMZ)
```

Defining unique CONNX environment variables per Started Task/Batch Job in the same CNXCFG VSAM KSDS is not supported. To do so, define separate copies of the CNXCFG VSAM KSDS for each started task/batch job, and update the execution JCL to point to the new CNXCFG VSAM KSDS. For example, to override the default CNXCFG DDNAME in the execution JCL:

```
//CNXCFG DD DSN=CONNx.STASK.CNXCFG.NEW,DISP=SHR
```

The REXX exec file CNXCFG supports overriding the CNXCFG VSAM KSDS file name via the following syntax:

From the ISPF Command Shell (usually ISPF main menu option 6):

Implicit:

```
cnxcfg p(variable value) f(CNXCFG data set name)
```

Verbose:

```
exec 'CONNx.STASK.CNTL(CNXCFG)' 'p(variable value) f(CNXCFG data set name)'
```

From any ISPF command line:

Implicit:

```
tso cnxcfg p(variable value) f(CNXCFG data set name)
```

Verbose:

```
tso exec 'CONNx.STASK.CNTL(CNXCFG)' 'p(variable value) f(CNXCFG data set name)'
```

2. For IDCAMS DELETE / DEFINE control records and execute JCL for the CNXCFG VSAM KSDS, refer to members IDCCNX0, CNXIDC, and @INST of CONNX.STASK.CNTL.

CONNX Environment Variable Security

Environment Variable Security vs. Ease of Use

You can implement CONNX environment variables exclusively by entering variable/value pairs in a CNXPARMS partitioned data set member, or via the CNXCFG REXX command file CNXCFG and program CNXCFGB to update the CNXCFG VSAM KSDS, or by a combination of both methods. Defining CONNX environment variables via the CNXPARMS PDS member is easier than using the REXX exec file, but less secure, depending on the security rules protecting the CONNX.STASK.CNTL partitioned data set. Defining the CONNX environment variables in the CNXCFG KSDS provides an extra layer of protection, because the TSO/ISPF editor cannot view/update VSAM data sets.

Part 4: Starting Multiple Instances of CONNX Started Tasks

Starting multiple instances of CONNX Started Tasks

Depending on the development needs and/or security rules defined at your site, it may be advantageous to define and start multiple instances of the CONNX for VSAM / QSAM / PDS server, either as started tasks, batch jobs, or a mixture of the two. Running the CONNX for VSAM / QSAM / PDS server as a batch job permits each user to test in a separate 'sandbox' environment, with separate versions of target VSAM / QSAM / PDS (physical sequential and partitioned data set) files, and without the need for systems programmer intervention to define new RACF user IDs for the started task procedure names. To define additional started task or batch job environment, use the following steps:

Define the Unique CONNX Environment Variables

1. Review the environment variables defined in default control member CNXPARMS in CONNX.STASK.CNTL. Begin with CNXRUNPORT, which must be unique per instance of the CONNX for VSAM TCP/IP Listener programs, inclusive of all currently running and regularly-scheduled CICS and Started Task versions of CNXRUN and CNXRUNB. Make a copy of this member for the new instance.
2. Review the persistent CONNX environment variables defined in the CNXCFG VSAM KSDS. Display the current persistent CONNX environment variables which reside in the default CNXCFG KSDS by executing the CNXCFG REXX exec file with no parameters.

From the ISPF Command Shell:

```
cnxcfg
```

or

```
exec 'connx.stask.cntl(cnxcfg)'
```

From any ISPF Command line:

```
tso cnxcfg
```

or

```
tso exec 'connx.stask.cntl(cnxcfg)'
```

3. The current variables/values defined in the default CNXCFG VSAM KSDS file are displayed on the invoking terminal:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu
Option ==> tso cnxcfg p(cnxrunport)

0 Settings      Terminal and user parameters      User ID . : IBMUSER
1 View         Display source data or listings   Time. . . : 19:09
2 Edit         Create or change source data   Terminal. : 3278
3 Utilities     Perform utility functions       Screen. . : 1
4 Foreground   Interactive language processing   Language. : ENGLISH
5 Batch        Submit job for language processing   Appl ID . : ISR
6 Command      Enter TSO or Workstation commands   TSO logon : ISPFPROC
7 Dialog Test  Perform dialog testing                   TSO prefix:
8 LM Facility  Library administrator functions       System ID : P390
9 IBM Products IBM program development products   MVS acct. : ACCT#
10 SCLM        SW Configuration Library Manager   Release . : ISPF 4.8
11 Workplace   ISPF Object/Action Workplace
M More         Additional IBM Products

Enter X to Terminate using log/list defaults
Parms = 'CNXRUNPORT'
loadlib = 'CONNX.STASK.load'
file = 'CONNX.STASK.cnxcfg'
***

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

file = 'CONNX.STASK.cnxcfg'
CONNX 8.90 (build 21)
Copyright (c) CONNX Solutions Inc. 1994-2003
(CNXCONNECTBACK)=(0)
(CNXNOPREAUTHORIZE)=(0)
(CNXRUNPORT)=(6710)
(DEBUG)=(0)
(FASTPATHMATCH)=(0)
(IMPORTALTINDEXES)=(0)
***

```

4. Decide which, if any, CONNX environment variables to define in the CNXCFG KSDS. If this support is required, define a new CNXCFG KSDS using members CNXIDC and IDCCNX0 of CONNX.STASK.CNTL, and update the CNXVSAM / CNXVSCT JCL procedures to point the CNXCFG

DDNAME to the new physical data set. If all CONNX environment variables are defined in the CNXPARMS PDS member, the CNXCFG DDNAME / DSN can be DUMMYed out, for example:

```
//CNXRUNB.CNXCFG DD DUMMY
```

5. To display, add, update, and delete environment variables/values defined in a new CNXCFG VSAM KSDS, you can use the f(file name) command-line parameter for the CNXCFG REXX file, for example:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu
Option ==> tso cnxcfg f(connx.v08r09.stask.cnxcfg)

0 Settings      Terminal and user parameters      User ID . : IBMUSER
1 View          Display source data or listings      Time. . . : 19:09
2 Edit          Create or change source data        Terminal. : 3278
3 Utilities      Perform utility functions             Screen. . : 1
4 Foreground     Interactive language processing       Language. : ENGLISH
5 Batch          Submit job for language processing     Appl ID . : ISR
6 Command        Enter TSO or Workstation commands       TSO logon : ISPFPROC
7 Dialog Test    Perform dialog testing                TSO prefix:
8 LM Facility     Library administrator functions      System ID : P390
9 IBM Products   IBM program development products     MVS acct. : ACCT#
10 SCLM          SW Configuration Library Manager     Release . : ISPF 4.8
11 Workplace     ISPF Object/Action Workplace
M More          Additional IBM Products

Enter X to Terminate using log/list defaults

*--- CONNX for VSAM Current Configuration: -----*
loadlib = 'CONNX.V08R09.STASK.load'
***

```



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
file = 'CONNX.V08R09.STASK.CNXCFC'
CONNX 8.90 (build 21)
Copyright (c) CONNX Solutions Inc. 1994-2003
(CNXCONNECTBACK)=0
(CNXNOPREAUTHORIZE)=0
(CNXRUNPORT)=6710
(DEBUG)=0
(FASTPATHMATCH)=0
(IMPORTALTINDEXES)=0
***
MA a 10/006

```

6. Alternatively, make a copy of the CNXCFC REXX exec file, and change the default value for the file parameter in the command text;

```

Session A - [24 x 80]
File Edit View Confirm Menu Utilities Compilers Test Help
EDIT CONNX.STASK.CNTL(CNXCFC) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 /* REXX */
000002 parse arg args
000003
000004 parms = ""
000005 file = ""
000006 loadlib = ""
000007 help = ""
000008 cpref = "CONNX.V08R09.STASK"
000009 lpref = "CONNX.V08R09.STASK"
000010 fpref = "CONNX.V08R09.STASK"
000011 progname = "cnxcfgb"
000012 instlib = lpref || ".load"
000013 cfgfile = ".cnxcfg"
000014
000015 do while (args ~= "")
000016   args = strip(args,L)
000017   parse value args with keyword "(" args
000018   parse value args with val ")" args
000019   saved_keyword = keyword

```


7. Change

fpref = "CONNX.STASK"

and

cfgfile = ".cnxcfg"

These entries are concatenated to form the CNXCFG physical data set name, which is by default CONNX.STASK.CNXCFG.

8. Copy or update the CNXVSAM / CNXVSCT JCL procedure(s). Review the VSAM DDNAMEs defined in the CNXVSAM/CNXVSCT JCL procedure(s), add your installation VSAM DDNAME/DSNs, and delete those DDNAMEs/DSNs which are not defined at your site. For newly added VSAM DDNAMEs /DSNs, follow the CONNX import procedures in "To test CONNX for VSAM Started Task / Batch Job SCT Meta Data Import Functionality" on page 5-113 to add VSAM metadata to the CONNX Data Dictionary file.

9. Decide whether to run the copied or updated CNXVSAM or CNXVSCT procedure(s) as started tasks or as batch jobs.

10. If new started task JCL procedures are created, ask your system programmer to define the started task IDs to RACF or a RACF-compatible security package. If the new CONNX environments are defined as batch jobs, make copies of the CNXVSAMJ or CNXVSCTJ JCL members in the CONNX.STASK.CNTL partitioned data set, and make the necessary changes to the job card and //EXEC statements to reference the new JCL procedure(s):

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Confirm Menu Utilities Compilers Test Help

EDIT          CONNX.STASK.CNTL(CNXVSAMJ) - 01.00          Columns 00001 00072
Command ==>                                     Scroll ==> CSR
000019 // JCLLIB   ORDER=CONNX.V08R09.STASK.CNTL
000020 //CNXVSAM   EXEC CNXVSAM ,CNXHLQ='CONNX.STASK',
000021 //***          CNXSAMP='CONNX.STASK.SAMPLES',
000022 //***          CNXPARMS=CNXPARMS,
000023 //***          SYSMOD='CONNX.STASK.LOAD',
000024 //***          TCPHLQ=TCPIP,
000025 //***          REG='8M'
000026 //
***** ***** Bottom of Data *****

```

MA a 04/015

11. Start the new task(s) or submit the new batch job(s) to the batch execution job queue.

Part 5: Uninstallation Procedures: CONNX Started Task Components

Introduction to uninstallation of CONNX Started Task components

The CONNX for VSAM TCP/IP Server Setup client application transfers the CONNX components to the target MVS-OS/390 host via TCP FTP. After the initial FTP step, the CONNX installation continues via a batch job submission using TSO ISPF. Final installation steps for the CONNX for VSAM Started Task components include manual additions of JCL procedures (CNXVSAM and CNXVSCT) and a REXX exec file (CNXCFG) to the system PROCLIB and SYSPROC partitioned data sets.

Given a default CONNX installation prefix of CONNX.V08R09.STASK, the CONNX for VSAM Started Task installation JCL partitioned data set is CONNX.V08R09.STASK.CNTL. Members REMOVE and REMOV01 contain batch uninstallation JCL. Use REMOVE if you installed the CONNX sample VSAM / QSAM / PDS data sets; otherwise, use REMOV01.

To uninstall the CONNX for VSAM Started Task components

1. Before submitting the uninstallation JCL, find and stop all instances of the CONNX for VSAM Started Task/Batch Job TCP/IP Listener program CNXRUNB. For example, three batch jobs are submitted to the batch execution queue, listening on TCP/IP ports 6710, 6720, and 6730:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
-----
Display Filter View Print Options Help
-----
SDSF DA SYS1 P390 PAG 0 SID 7 CPU 16 DATA SET DISPLAYED
COMMAND INPUT ==> SCROLL ==> CSR
PREFIX=CNX* DEST=(ALL) OWNER=* SORT=JOBNAME/A
NP JOBNAM STEPNAME PROCSTEP JOBJID OWNER C POS DP PGN REAL PAGING S
CNXVSAM1 CNXVSAM CNXRUNB JOB07030 IBMUSER A LO FF 1 446 0.00 0.
CNXVSAM2 CNXVSAM CNXRUNB JOB07031 IBMUSER A LO FF 1 483 0.00 0.
CNXVSAM3 CNXVSAM CNXRUNB JOB07032 IBMUSER A LO FF 1 459 0.00 0.
-----
MA a 10/002

```

2. The TCP/IP ports dedicated to these jobs can be found via the TSO NETSTAT command:

```

EZZ2587I CNX VSAM1 00002810 0.0.0.0..6710 0.0.0.0..0
Listen

```

```

EZZ2587I CNXVSAM2 00002812 0.0.0.0..6720 0.0.0.0..0
Listen

```

```

EZZ2587I CNXVSAM3 00002813 0.0.0.0..6730 0.0.0.0..0
Listen

```

3. In the current case, there are no active client connections to the three jobs, which are in the idle Listen state.
4. Execute the CNXCFG REXX command file using either the implicit or the verbose syntax below, and repeat the command for ports 6720 and 6730:

```

From the ISPF Command Shell:
cnxcfg p(stop 6710)
or
exec 'connx.stask.cntl(cnxcfg)' 'p(stop 6710)'
From any ISPF Command line:
tso cnxcfg p(stop 6710)
or

```

```
tso exec 'connx.v08r09.stask.cntl(cnxcfg)' 'p(stop 6710)'
```

5. Repeat this process for each instance of the CNXVSAM/CNXSCT Started Task(s)/Batch Job(s).
6. After stopping all instances of the CNXVSAM / CNXVSCT Started Task(s)/Batch Job(s), you can submit the JCL in member @REMOVE or @REMOV01. Note that the last control record of the IDCAMS DELETE step references the CONNX.STASK.CNTL partitioned data set, so be sure to submit the job and deallocate the partitioned data set from your TSO session by navigating to another ISPF option:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Confirm Menu Utilities Compilers Test Help

EDIT      CONNX.STASK.CNTL(@REMOV10) - 01.00      Columns 00001 00072
Command ==>                                     Scroll ==> CSR
000080  DELETE CONNX.STASK.SAMPLES.ORDER.R      PURGE SCRATCH
000081  DELETE CONNX.STASK.SAMPLES.PRODGRP      PURGE SCRATCH
000082  DELETE CONNX.STASK.SAMPLES.PRODGRP.E    PURGE SCRATCH
000083  DELETE CONNX.STASK.SAMPLES.PRODGRP.PDS  PURGE SCRATCH
000084  DELETE CONNX.STASK.SAMPLES.PRODGRP.PS   PURGE SCRATCH
000085  DELETE CONNX.STASK.SAMPLES.PRODGRP.R    PURGE SCRATCH
000086  DELETE CONNX.STASK.SAMPLES.PRODUCT      PURGE SCRATCH
000087  DELETE CONNX.STASK.SAMPLES.PRODUCT.E    PURGE SCRATCH
000088  DELETE CONNX.STASK.SAMPLES.PRODUCT.PDS  PURGE SCRATCH
000089  DELETE CONNX.STASK.SAMPLES.PRODUCT.PS   PURGE SCRATCH
000090  DELETE CONNX.STASK.SAMPLES.PRODUCT.R    PURGE SCRATCH
000091  DELETE CONNX.STASK.CNTL                PURGE SCRATCH
000092  /*
000093  /**STEP2      EXEC PGM=IDCAMS
000094  /**SYSPRINT DD SYSOUT=*
000095  /**SYSIN      DD *
000096  /**DELETE CONNX.STASK.LOAD      PURGE SCRATCH
000097  /**
***** ***** Bottom of Data *****

```

7. Also note that the second IDCAMS DELETE step for the CONNX.STASK.LOAD partitioned data set is commented out. This is by design, in case the CONNX Started Task/Batch job load modules were linked into an existing load library. It is recommended, but not required, to install CONNX into a separate load library. If the CONNX load modules were installed into a new load library:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Functions Confirm Utilities Help

DSLIST          CONNX.STASK.LOAD          Row 00001 of 00003
Command ==>          Scroll ==> CSR
      Name      Prompt      Alias-of      Size      TTR      AC      AM      RM
      -----
      CNXCFGB          000C2090      001809      00      31      24
      CNXRUNB          000C97C0      000007      01      31      24
      CNXVSAMB          000EDC18      003104      01      31      24
      **End**

. . . . .
MA a 04/015

```

8. Then Step 2 can be commented in:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Confirm Menu Utilities Compilers Test Help

EDIT          CONNX.STASK.CNTL(@REMOV10) - 01.00          Columns 00001 00072
Command ==>          Scroll ==> CSR
000080  DELETE CONNX.STASK.SAMPLES.ORDER.R          PURGE SCRATCH
000081  DELETE CONNX.STASK.SAMPLES.PRODGRP          PURGE SCRATCH
000082  DELETE CONNX.STASK.SAMPLES.PRODGRP.E          PURGE SCRATCH
000083  DELETE CONNX.STASK.SAMPLES.PRODGRP.PDS          PURGE SCRATCH
000084  DELETE CONNX.STASK.SAMPLES.PRODGRP.PS          PURGE SCRATCH
000085  DELETE CONNX.STASK.SAMPLES.PRODGRP.R          PURGE SCRATCH
000086  DELETE CONNX.STASK.SAMPLES.PRODUCT          PURGE SCRATCH
000087  DELETE CONNX.STASK.SAMPLES.PRODUCT.E          PURGE SCRATCH
000088  DELETE CONNX.STASK.SAMPLES.PRODUCT.PDS          PURGE SCRATCH
000089  DELETE CONNX.STASK.SAMPLES.PRODUCT.PS          PURGE SCRATCH
000090  DELETE CONNX.STASK.SAMPLES.PRODUCT.R          PURGE SCRATCH
000091  DELETE CONNX.STASK.CNTL          PURGE SCRATCH
000092  /*
000093  //STEP2      EXEC PGM=IDCAMS
000094  //SYSPRINT DD SYSOUT=*
000095  //SYSIN      DD *
000096  DELETE CONNX.STASK.LOAD          PURGE SCRATCH
000097  //
***** ***** Bottom of Data *****
. . . . .
MA a 04/015

```

9. The IDCAMS DELETE steps should return zeroes for all data sets which are not allocated to another batch job, started task, or TSO session. For data sets which do not exist or are allocated to other tasks, the IDCAMS return code will be 8. The undeleted data sets, if any, can be deleted via ISPF or via a second run of the @REMOVE or @REMOV01 JCL after the data set sharing conflicts have been resolved.

Manual Steps: CONNX for VSAM Started Task Uninstallation

To verify that all CONNX for VSAM Started Task components have been uninstalled, you can use ISPF option 3.4 to list and delete the data sets defined with the (default) CONNX.STASK high-level qualifier. Next, delete all versions of the CNXVSAM/CNXVSCT procedures from the production JCL PROCLIB. Finally, if the CNXCFG REXX exec file has been copied to a SYSPROC PDS such as SYS1.LOCAL.CLIST, delete the CNXCFG member.

Dataset Security

CONNX VSAM / QSAM / PDS Host / Client Security Overview

Default Option - Figure #1:

CONNX for VSAM / QSAM / PDS is implemented as a Windows PC client and a set of batch programs, including a Listener (CNXRUNB) and a Server (CNXVSD0B). When a user connects from a client application through a supported interface (ODBC, JDBC, OLE DB or .NET), the client CONNX user ID / password is mapped to an OS/390 / z/OS user ID / password via the CONNX Data Dictionary.

The encrypted user ID / password is sent via TCP/IP to the CONNX TCP/IP Listener program CNXRUNB, which decrypts the user ID / password and executes the RACF (Resource Access Control Facility) VERIFY macro. If the user ID / password sent from the CONNX PC client is valid, program CNXRUNB starts program CNXVSD0B as a subtask in the CNXRUNB started task or batch job address space.

Refer to steps 1 through 3 in Figure 1. Once started, program CNXVSD0B sets up a separate TCP/IP connection to the invoking CONNX PC client.

CONNX PC Client requests and VSAM / QSAM / PDS data responses flow back and forth directly from the CONNX PC client to the dedicated CNXVSD0B subtask. Refer to steps 4 and 5 in Figure 1. For the default case, the host-side RACF dataset security rules defined for the CNXVSD0B subtask user IDs (#1, #2, or #3) determine the type of file access granted to each CONNX PC client.

The default security option observes the host-side security rules defined for user ID / password verification, and per-user ID for VSAM / QSAM / PDS file access. In order for these rules to be enforced, programs CNXRUNB and CNXVSD0B must run from an APF (Authorized Program Facility) load library.

The operator command (SETPROG APF) necessary to add the CONNX started task load library to the APF list is documented in the CONNX Installation Guide.

Alternate Option - Figure #2:

The host-side security rules for user ID / password verification and dataset access can be enabled or disabled for CONNX client-server connections via a CONNX environment variable (CNXNOPREAUTHORIZE). Setting CNXNOPREAUTHORIZE to a non-zero value instructs the CONNX VSAM / QSAM / PDS TCP/IP Listener and Server programs to bypass user ID / password verification and file access security checks. Once the listener program (CNXRUNB) starts a server program (CNXVSD0B) subtask, client requests and VSAM data flow from the CONNX PC client to the server subtask and back via a dedicated TCP/IP socket connection. In this case, the host-side security rules defined for the CNXRUNB user ID (#0) attach to each CNXVSD0B server subtask, and determine the type of file access granted to the CONNX PC client.

This security option bypasses the need to execute the CONNX programs from an APF-authorized load library, but all dataset access derives from the single user ID which executes listener program CNXRUNB as a started task or a batch job. An advantage to this approach is that host-side data set security rules for CONNX PC clients need only be defined for each user ID which starts program CNXRUNB.

CONNX Client-Side Security Enhances Host-Side Security

For both options, the CONNX Administrator should consider taking advantage of the client-side security features implemented in the CONNX Data Dictionary (CDD). The first line of defense is only to import selected VSAM / QSAM / PDS files into the CDD. Additionally, the CONNX Administrator can restrict file access based on CONNX user IDs and groups defined via the security menu features in the CONNX Data Dictionary Manager.

In some cases, a physical VSAM file is composed of multiple logical files or sub-files. The CONNX Data Dictionary Manager can be used to define and enforce security rules based on these logical files, as well as on the underlying physical data sets. Further, column- and row-level security can be implemented by defining one or more CONNX views against the imported VSAM physical or logical files, and by authorizing individual or groups of CONNX user IDs to execute the CONNX client-side views.

Similarly, host-side dataset security rules for PDS (partitioned data set) files are implemented at the dataset level. CONNX client-side security can restrict file access per-PDS member, as well as to specific columns or rows within a PDS member.

The CONNX client-side approach to security complements and enhances most host-side security products, which implement file access rules on a per-physical file basis. In all cases, the security rules defined in the CONNX Data Dictionary take precedence over the host-side security rules. For more information on CONNX client-side security, refer to the CONNX online User Reference Guide.

Figure 1

CONNX VSAM / QSAM / PDS Default Security

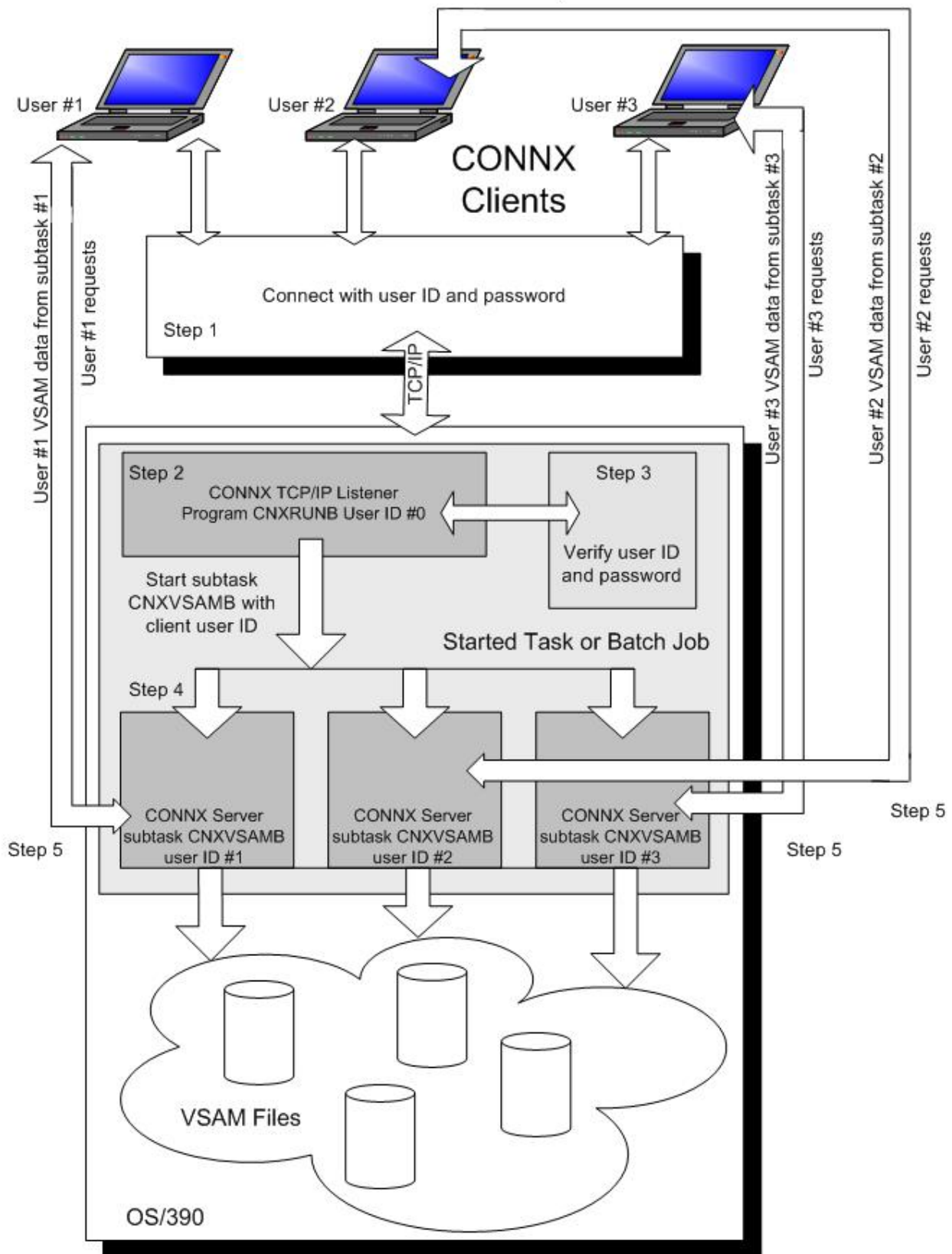
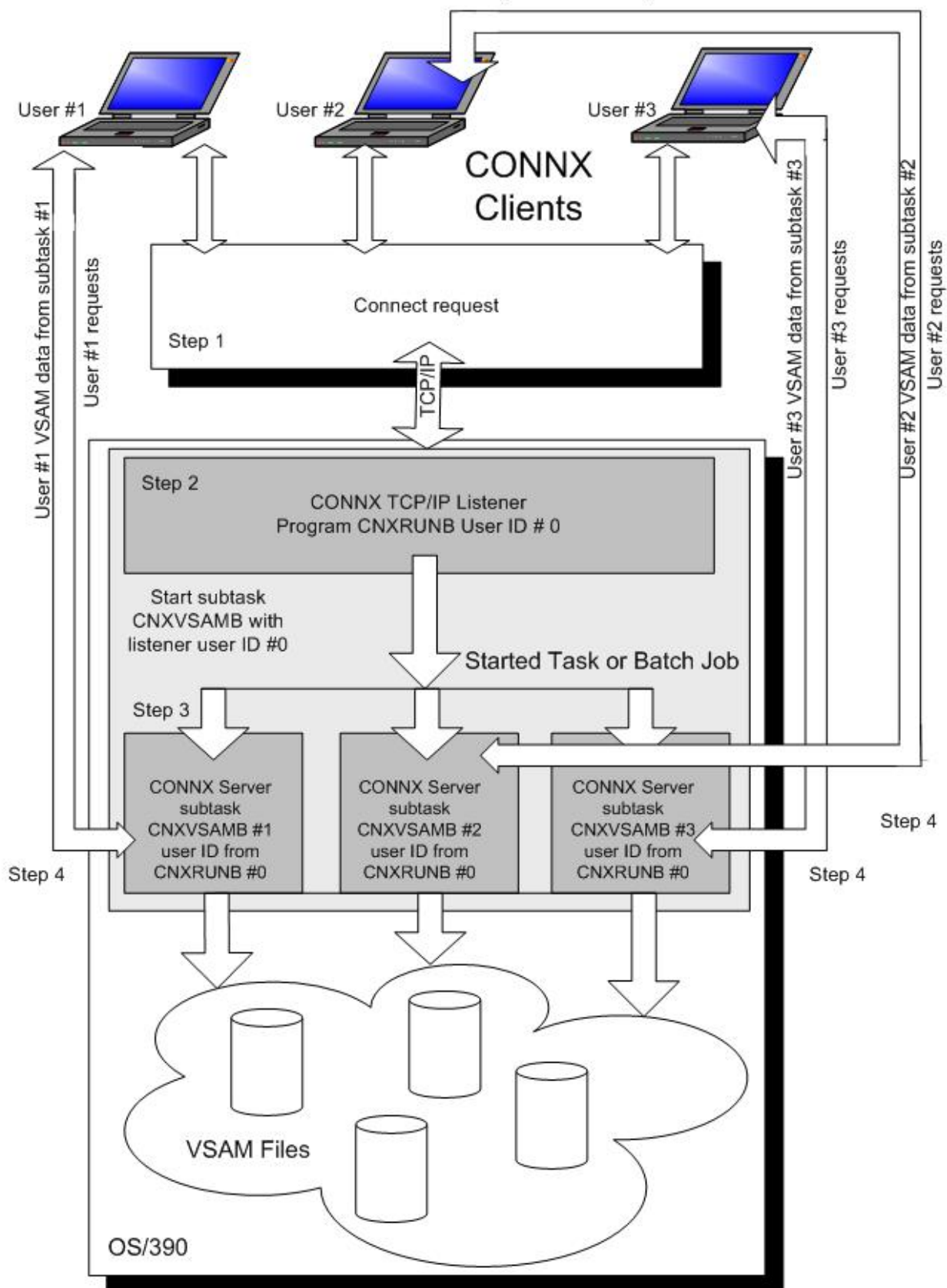


Figure 2

CONNX VSAM / QSAM / PDS Client-side Only Security



CONNX for VSAM - CICS

Prerequisites - CONNX for CICS/VSAM - z/OS

- The IBM z/OS mainframe must be set up for TCP/IP communications.
- The TCP/IP started task must be running.
- The IBM CICS TCP/IP sockets interface, including CICS configuration files, programs, maps, and transactions, must be installed.
- Both the install PC client and the target host (z/OS) should have FTP enabled (optional).

Refer to IBM documentation for more information on installing the IBM CICS TCP/IP interface.

Verifying TCP/IP Connectivity

Before you install CONNX on your CICS/VSAM server, we recommend that you verify the TCP/IP connectivity to your OS/390 z/OS system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

ping hostname

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Part 1: Installing the CONNX VSAM Components on an MVS-OS/390 Server

General Information about the CONNX VSAM Server

This step copies the CONNX VSAM server components from a client PC to an OS/390 / z/OS target. The CONNX Administrator component must be installed on the client PC.

About the VSAM User Account and Password

When installing the VSAM components, select a TSO user account name and password with privileges that allow you to create partitioned data sets and to copy, read, and write files to the server. The installation user ID must be authorized to create partitioned data sets and to copy files to the target host via FTP.

To install the CONNX components on the target CICS/VSAM z/OS system

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears.

2. Select the OS/390 / z/OS tab in the CONNX Server Component Installation dialog box. Select the VSAM option in the **Product** area. Select the CICS option in the **Server Type** area.

Note: Additional information about the information in these dialog boxes can be found in the **Required and Optional Entry Fields for VSAM TCP/IP Server Setup** table.

3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your OS/390 / z/OS server in the **Server** text box, a TSO user ID in the **User ID** text box and its password in the **Password** text boxes.
4. In the **TCP/IP area**, enter the dataset name of the TCP/IP stack in the **HLQ** text box and the TCP/IP library name in the **Library** text box.
5. In the **CONNX Server Data Set** area, enter the data set name of the CONNX server component location (where CONNX for VSAM will be installed on the mainframe) in the **CONNX DSN HQL** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text box. If SMS is installed on the CONNX server system, select the **SMS** check box.
6. In the **CICS Parameters** area, enter the data set name of the link step output files in the **Load Library DSN** text box, its VOLSER in the **VOLSER** text box, and its DASD device type in the **UNIT** text

box. Enter the low level nodes of the CICS region load library in **Product Load Library** and the Resource Management Commands file in **Target RDO File**.

7. In the **VSAM Samples** area, select the **Create** check box if you want to create VSAM sample files. Enter the data set name of the CONNX sample file location in **Database HLQ** and its VOLSER in the **VOLSER** text boxes. If the data set prefix is CONNX.STASK, this will create the VSAM, QSAM, and partitioned data sets listed in [VSAM Objects for CONNX.STASK.SAMPLES](#) and load them with sample data.

If you do not want sample files, clear the **Create** check box.

8. In the **FTP Mode area**, enter the FTP mode that is appropriate for your FTP server. If the host cannot be accessed via FTP, select Manual Copy. Manual Copy will create a local copy of the files that need to be transferred along with a text file with instructions.

9. Click the **Install** button. This action starts an FTP session and copies the selected components from the client PC to the target host.

10. Allow several minutes for the file transfer process between the CONNX administrator computer directory C:\CONN32\VSAM and the target host.

11. Click the **Close** button.

Required and Optional Entry Fields for VSAM TCP/IP Server Setup

Group Box	Term	Description
Login	CICS/VSAM Server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.012
	User ID	Your 1-8 character z/OS or TSO logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character z/OS or TSO password. All characters convert from lowercase to uppercase.
Server Type	CICS	The default installation for CONNX for CICS/VSAM on OS/390 / z/OS platforms is as a CICS TCP/IP listener and server.
	Started Task	CONNX for VSAM can also be installed and run as one or more started tasks or batch jobs. This version also provides read access to QSAM physical sequential and partitioned data sets.
CONNX Server	CONNX DSN Prefix	A one- or multi-part high-level data set prefix, which is

Data Set		used to create the CONNX installation sequential and partitioned data sets on the target host. In the current example, the installation prefix is CONNX.CICS. See CONNX DSN Prefix Data Sets for a description of the data sets created with this prefix.
	VOLSER	Optional. The DASD volume serial on which the CONNX.CICS sequential and partitioned data sets are created.
	UNIT	Optional. The DASD device type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.
VSAM Samples	VSAM Samples Prefix	Required if the Create check box in the VSAM Samples group box is checked. A one- or multi-part high-level data set prefix, which is used to create the CONNX VSAM sample data sets. See VSAM Objects for CONNX.CICS.SAMPLES and VSAM Objects for CONNX.STASK.SAMPLES for lists of the sample VSAM files created for the target CIS region(s) or started task(s)/batch jobs.
	VOLSER	Required if the Create check box in the VSAM Samples group box is checked. The DASD volume serial on which the CONNX.CICS.SAMPLES VSAM objects are created.
CICS Parameters	CICS Product Load Library	Required for all three CICS installation options. The fully qualified CICS runtime load library partitioned data set name. For example, the CICS Transaction Server 1.3 default load library name is: CICSTS13.CICS.SDFHLOAD Ask your CICS administrator for the fully qualified name of this data set.
	Target RDO/File	Required for all three CICS installation options. The fully qualified name of the target CICS Resource Definition Online VSAM file. For example, the default name for the CICS Transaction Server 1.3 RDO file is: CICSTS13.CICSTS13.DFHCSD
	Link into Target Load Library	Required for both CONNX installation options. A fully qualified partitioned data set name into which the CONNX CICS/C++ object code is linked. This can be a new partitioned data set, or an existing PDS already defined to the target CICS region in the DFHRPL load list. In the current example, the target load library is CONNX.CICS.LOAD
	VOLSER.	The DASD volume serial on which the new CONNX load library is created. Specify a VOLSER and a UNIT to create the PDS on a specific volume, or leave both fields blank to accept the system defaults. If linking into an existing PDS, leave the VOLSER and UNIT text boxes blank.
	UNIT	Optional. The DASD device type (3380, 3390, etc.) of the above VOLSER. Omit if the VOLSER field is blank.

CONNX DSN Prefix Data Sets

Data Set	Description
CONNX.CICS.CNTL	Installation JCL and control records partitioned data set.
CONNX.CICS.COPYBOOK	Sample COBOL copybook partitioned data set.
CONNX.CICS.CUSTOMER.S	Sequential input data set used to load the sample CUSTOMER VSAM file. (CICS files CNXCSTE, CNXCSTK, CNXCSTR)
CONNX.CICS.EQUIPMNT.S	Sequential input data set used to load the sample EQUIPMENT VSAM files. (CICS files CNXEQE, CNXEQK, CNXEQR)
CONNX.CICS.OBJECT	CONNX SAS/C 7.00 pre-linked object code for the CONNX CICS and batch programs.
CONNX.CICS.ORDER.S	Sequential input data set used to load the sample ORDER VSAM file. (CICS files CNXPGE, CNXPBK, AND CNXPGR)
CONNX.CICS.PRODGRP.S	Sequential input data set used to load the sample Product Group VSAM file. (CICS file CNXPGRP)
CONNX.CICS.PRODUCT.S	Sequential input data set used to load the sample PRODUCT VSAM file. (CICS file CNXPROD)

VSAM Objects for CONNX.CICS.SAMPLES

VSAM Object	Description	CICS File Name	CONNX for VSAM Sample Table Name
CONNX.CICS.SAMPLES.CUSTOMER	Customer file KSDS.	CNXCSTK	CUSTOMERS_VSAM
CONNX.CICS.SAMPLES.CUSTOMER.E	Customer file ESDS.	CNXCSTE	CUSTOMERS_VSAM_ESDS
CONNX.CICS.SAMPLES.CUSTOMER.R	Customer file RRDS.	CNXCSTR	CUSTOMERS_VSAM_RRDS
CONNX.CICS.SAMPLES.EQPATH01	Equipment file path #1.	CNXEQP1	*none*
CONNX.CICS.SAMPLES.EQPATH01	Equipment file path #1.	CNXEQP2	*none*
CONNX.CICS.SAMPLES.EQUIPMNT	Equipment file KSDS.	CNXEQK	EQUIPMNT_VSAM
CONNX.CICS.SAMPLES.EQUIPMNT.E	Equipment file ESDS.	CNXEQE	EQUIPMNT_VSAM_ESDS
CONNX.CICS.SAMPLES.EQUIPMNT.R	Equipment file RRDS.	CNXEQR	EQUIPMNT_VSAM_RRDS
CONNX.CICS.SAMPLES.ORDER	Order file KSDS.	CNXORK	ORDERS_VSAM
CONNX.CICS.SAMPLES.ORDER.E	Order file ESDS.	CNXORE	ORDERS_VSAM_ESDS
CONNX.CICS.SAMPLES.ORDER.R	Order file RRDS.	CNXORR	ORDERS_VSAM_RRDS
CONNX.CICS.SAMPLES.ORDPTH01	Order file path #1.	CNXORP1	*none*
CONNX.CICS.SAMPLES.ORDPTH02	Order file path #2.	CNXORP2	*none*
CONNX.CICS.SAMPLES.PRDGRP	Product group file KSDS.	CNXPBK	PRODUCT_GROUPS_VSAM
CONNX.CICS.SAMPLES.PRODGRP.E	Product Group file ESDS.	CNXPGE	PRODUCT_GROUPS_VSAM_ESDS
CONNX.CICS.SAMPLES.PRODGRP.R	Product file RRDS.	CNXPGR	PRODUCTS_GROUPS_VSAM_RRDS
CONNX.CICS.SAMPLES.PRODUCT	Product file KSDS.	CNXPRDK	PRODUCTS_VSAM
CONNX.CICS.SAMPLES.PRODUCT.E	Product file ESDS	CNXPRDE	PRODUCTS_VSAM_ESDS

CONNX.CICS.SAMPLES.PRODUCT.R	Product file RRDS.	CNXPRD R	PRODUCTS_VSAM_RRDS
-------------------------------------	-----------------------	-------------	--------------------

Part 2: Installation of the CONNX components**To install the selected components to the OS/390 target**

1. Log on the target OS/390 system. Given the default data set prefix of CONNX.CICS, the ftp process creates a JCL control record partitioned data set with the name and members listed in the following table.
2. Edit the INST member and change the job card to match your site standards.
3. Submit this job to the batch execution queue.

Default Partitioned Data Set Name = CONNX.CICS.CNTL

Member	Purpose
INST	The CONNX and SAS/C 7.00 installation job built by the CONNX VSAM installation program.
INST02	The CONNX installation job created by the CONNX PC installation program. Submit this job to install CONNX onto a second CICS region.
REMOV0	CONNX for VSAM uninstallation JCL.
CICSALOC	JCL Procedure to allocate the SAS/C CICS transient data queues.
CICSDCT	Sample CICS Destination Control Table (DCT) source with entries for the IBM CICS TCP/IP interface and the SAS/C extra-partition transient data queues.
CICSDCTJ	Sample JCL to assemble / link-edit the CICSDCT source code into a target CICS load library.
CNXCSD	JCL procedure to execute IBM DFHCSDUP utility and define CONNX CICS components to a target RDO VSAM file.
CNXFCT0	FCT source for CONNX CICS/VSAM configuration and sample files.
CNXIDC	JCL procedure to execute IDCAMS to delete/define/load CONNX VSAM components.
CNXLINK	JCL procedure to execute the IBM linkage editor against CONNX prelinked SAS/C 7.00 object code and to create the CONNX CICS and batch load modules.
CNXLOAD	JCL procedure to delete/allocate a separate load library for the CONNX CICS programs.
CNXPPT0	PPT source for CONNX CICS/VSAM programs.
CSDFILE0	IBM DFHCSDUP utility control records to define the CONNX TCP/IP server VSAM configuration file to the target CICS RDO VSAM file.
CSDPROG0	IBM DFHCSDUP utility control records to define the CONNX CICS programs and transactions to the target CICS RDO VSAM file.
CSDSAMP0	IBM DFHCSDUP utility control records to define the CONNX CICS sample VSAM files, paths, and alternate indexes to the target CICS RDO VSAM file.
IDCCNX0	IDCAMS control records to drop the CONNX TCP/IP server VSAM configuration file.

IDCCUST0	IDCAMS control records to delete/define/load the CONNX sample VSAM customer file.
IDCEQU00	IDCAMS control records to delete/define/load the CONNX sample VSAM equipment file.
IDCEQU01	IDCAMS control records to define the CONNX sample VSAM equipment file alternate index #1.
IDCEQU02	IDCAMS control records to define the CONNX sample VSAM equipment file alternate index #2.
IDCEQU03	IDCAMS control records to build the CONNX sample VSAM equipment file alternate index #1.
IDCEQU04	IDCAMS control records to build the CONNX sample VSAM equipment file alternate index #2.
IDCORD00	IDCAMS control records to delete/define/load the CONNX sample VSAM order file.
IDCORD01	IDCAMS control records to define the CONNX sample VSAM order file alternate index #1.
IDCORD02	IDCAMS control records to define the CONNX sample VSAM order file alternate index #2.
IDCORD03	IDCAMS control records to build the CONNX sample VSAM order file alternate index #1.
IDCORD04	IDCAMS control records to build the CONNX sample VSAM order file alternate index #2.
IDCPGRP0	IDCAMS control records to delete/define/load the CONNX sample VSAM product group file.
IDCPROD0	IDCAMS control records to delete/define/load the CONNX sample VSAM product file.
LNKCFG	IBM linkage editor control records for CONNX CICS load module CNXCFG.
LNKRUN	IBM linkage editor control records for CONNX CICS load module CNXRUN.
LNKSTOP	IBM linkage editor control records for CONNX load module CNXSTOP.
LNKVSAM	IBM linkage editor control records for CONNX CICS load module CNXVSD0.

If you requested that a new CONNX load library be created by specifying a new PDS name in the Target Load Library text box of the installation program, then the example CONNX CICS load library is allocated. In both cases, the CONNX load module members are linked into the Target Load Library PDS:

CONNX.CICS.LOAD

CNXCFG
CNXRUN
CNXVSD0
CNXSTOP

Completing CICS Customization

After the required CONNX CICS components are installed, they must be defined to the target CICS region. The CONNX installation automates the definition of CONNX CICS programs, files, and transactions to the target region RDO file. Depending on the CICS version and release installed on the target region, optional manual steps include updating the DCT (Destination Control Table), the PPT (Program Processing Table), and PCT (Program Control Table). If the CONNX load modules were linked into a new load library, then the final required manual step is to add the CONNX load library (default name = CONNX.CICS.LOAD) to the target CICS region startup JCL DFHRPL (Run Program List) DDNAME.

To change the CICS startup JCL RPL list

- Concatenate the CONNX load library partitioned data set name to the CIS startup JCL DFHRPL DDNAME:

Example:

The screenshot shows a mainframe terminal window titled "Session B - [24 x 80]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. Below the menu bar, there is a sub-menu bar with File, Edit, Confirm, Menu, Utilities, Compilers, Test, and Help. The main display area shows the following text:

```

EDIT          SYS1.PROCLIB(CICSTEST) - 01.99          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
000079 //DFHRPL      DD DSN=CONNX.CICS.SYSTEM.LOAD, <== DCT,PCT,PPT etc.
000080 //              DCB=BLKSIZE=32760,
000081 //              DISP=SHR
000082 //              DD DSN=CONNX.CICS.LOAD, CONNX CICS load modules
000083 //              DISP=SHR
000084 //              DD DSN=&INDEX2..SDFHLOAD,          <== IBM CICS runtime
000085 //              DISP=SHR
000086 //              DD DSN=&INDEX2..SDFHAUTH,
000087 //              DISP=SHR
000088 //              DD DSN=TCPIP.SEZATCP,          <== OS/390 TCP/IP runtime
000089 //              DISP=SHR
000090 //              DD DSN=CEE.SCEECICS,          <== LE/390 CICS runtime
000091 //              DISP=SHR
000092 //              DD DSN=CEE.SCEERUN,          <== LE/390 runtime
000093 //              DISP=SHR
000094 //              DD DSN=CICSTS13.CICS.SDFHSAMP, <== IBM CICS Sample Apps
000095 //              DISP=SHR
000096 // *
000097 // *
000098 // *
  
```

At the bottom left of the window, there is a status bar showing "MA" and "b". At the bottom right, it shows "04/015".

Installation Verification of CONNX CICS Components

Option 1 - RDO: If you request to install the CONNX CICS components, the installation job executes the IBM DFHCSDUP utility to define the CONNX programs, transactions, and VSAM configuration and sample files to the target CICS partition RDO VSAM file. You can verify the installation of the CONNX CICS components by logging on to the target CICS partition and issuing the following command:

```
CEDA EXPAND GR(CNXVVRR)
```



```

EXPAND GR(CNXVVRR)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
CNXCFG    FILE      CNXVVRR    03.090    20.30.21
CNXCSTE    FILE      CNXVVRR    03.090    20.35.42
CNXCSTK    FILE      CNXVVRR    03.090    20.35.41
CNXCSTR    FILE      CNXVVRR    03.090    20.35.43
CNXEQE     FILE      CNXVVRR    03.090    20.35.53
CNXEQI1    FILE      CNXVVRR    03.090    20.35.54
CNXEQI2    FILE      CNXVVRR    03.090    20.35.55
CNXEQK     FILE      CNXVVRR    03.090    20.35.52
CNXEQP1    FILE      CNXVVRR    03.090    20.35.56
CNXEQP2    FILE      CNXVVRR    03.090    20.35.57
CNXEQR     FILE      CNXVVRR    03.090    20.35.53
CNXORE     FILE      CNXVVRR    03.090    20.35.50
CNXORI1    FILE      CNXVVRR    03.090    20.35.58
CNXORI2    FILE      CNXVVRR    03.090    20.35.59
CNXORK     FILE      CNXVVRR    03.090    20.35.49
CNXORP1    FILE      CNXVVRR    03.090    20.36.00
+ CNXORP2    FILE      CNXVVRR    03.090    20.36.01

RESULTS: 1 TO 17 OF 31
PF 1 HELP      3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

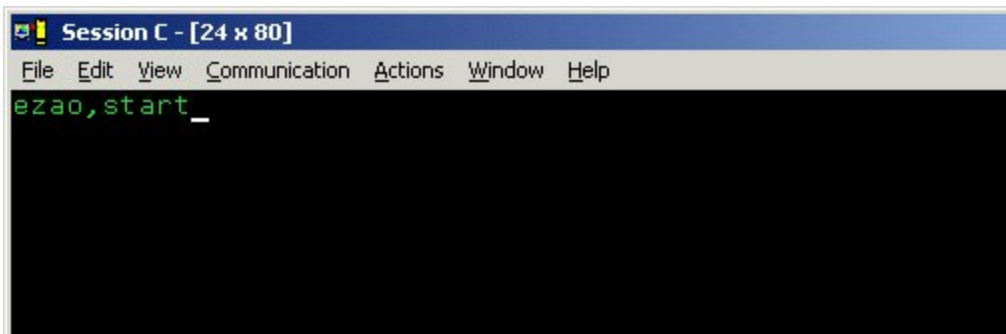
```

As needed, you can ask your CICS Administrator to add the CNXLIST list to the target CICS partition startup list. CNXLIST contains the CNXVVRR and SASC700 RDO groups.

Option 2 - Non-RDO: If the target CICS partition does not use RDO, you can define the CONNX CICS files and programs to the target FCT and PPT. The CONNX FCT entries are copied to member CNXFCT0 of the installation JCL partitioned data set; the PPT entries to member CNXPPT0. These entries must be manually copied to the master source for the CICS FCT and PPT and reassembled.


Part 3: CONNX Configuration Utility**To start the IBM CICS TCP/IP sockets interface - IMS**

1. Type **EZAO, START** and then press **<ENTER>**.



```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
ezao,start_
```

2. Type **Yes** after CICS to start the CICS TCP/IP sockets interface, and then press **<ENTER>**.

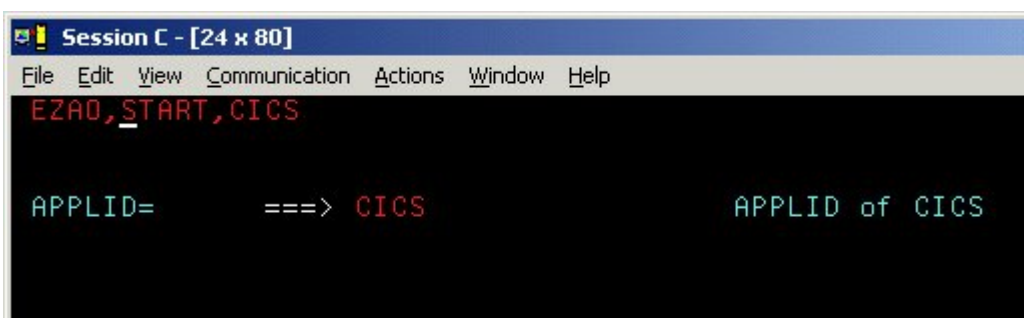


```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
EZAO,START
Enter one of the following

CICS          ==> yes_
LISTENER      ==>

Enter Yes|No
Enter Yes|No
```

3. The CICS APPLID appears after START.



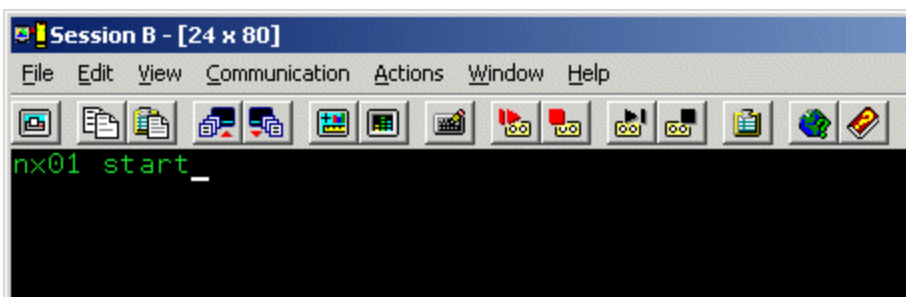
```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
EZAO,START,CICS

APPLID=       ==> CICS      APPLID of CICS
```

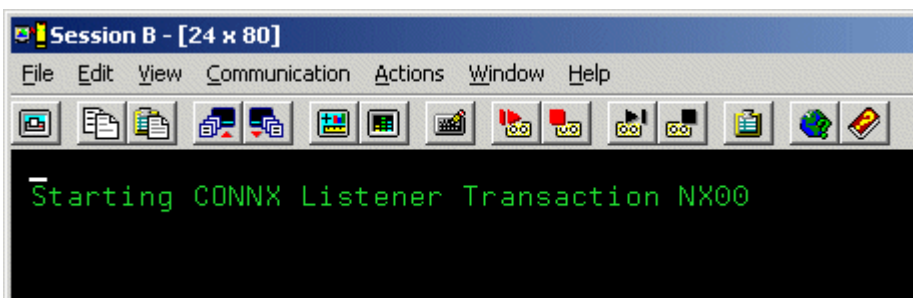
To start the CONNX CICS/VSAM TCP/IP Listener

The listener process waits for and verifies connections with client machines.

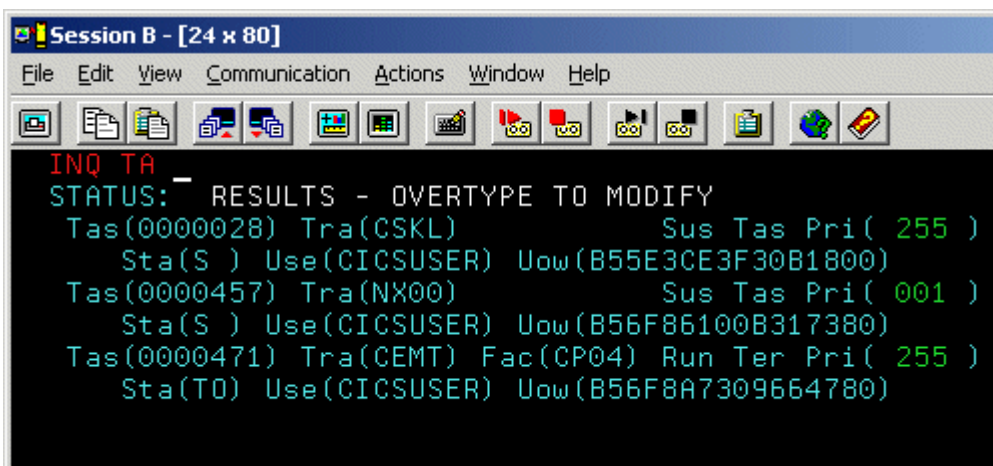
1. After logging on to the CICS region, enter **NX01 Start**, and then press **<Enter>**.



2. The CONNX TCP/IP Listener starts.

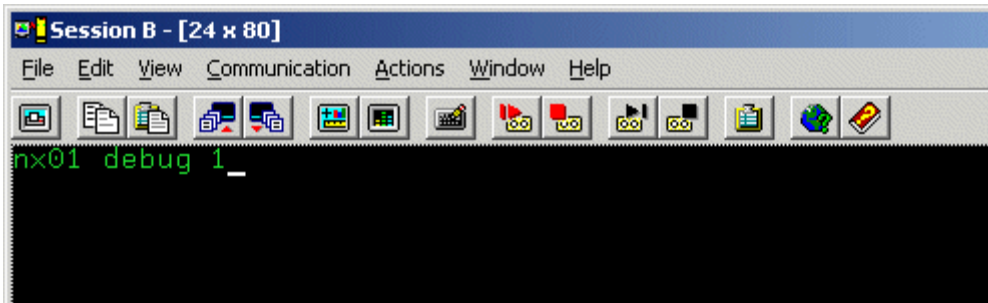


3. Verify that the CONNX CICS/VSAM TCP/IP Listener transaction NX00 started by typing **CEMT INQ TA**. Press **<Enter>**. The screen displays a list of transactions currently running. Verify that **Tas(0000xxx) Tra(NX00)** is listed as one of the transactions.

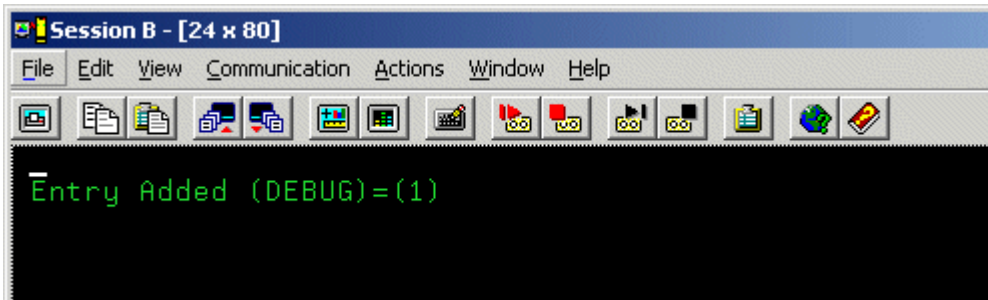


To set a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE> <VALUE>**, and then press **<Enter>** to turn on the CONNX variable function. In this example, the DEBUG variable is used.

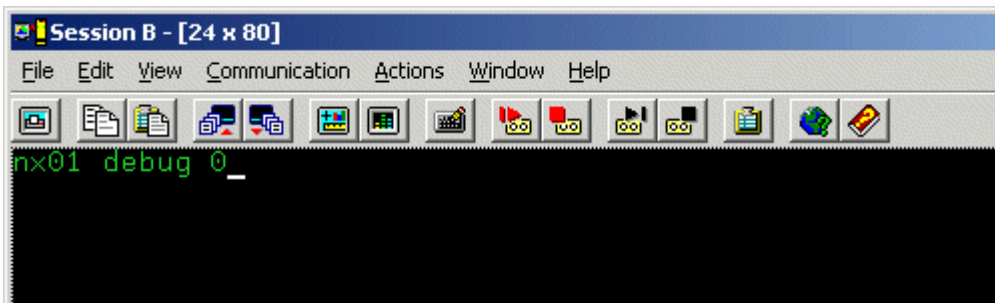


2. The variable is set, and verification appears.

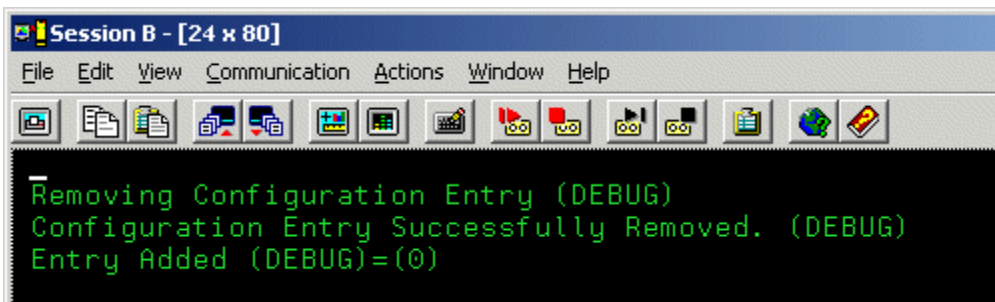


To change a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE>**, and then press **<Enter>**. In this example, in which the variable was set in [To set a CONNX environment variable](#), the **DEBUG** variable is used.

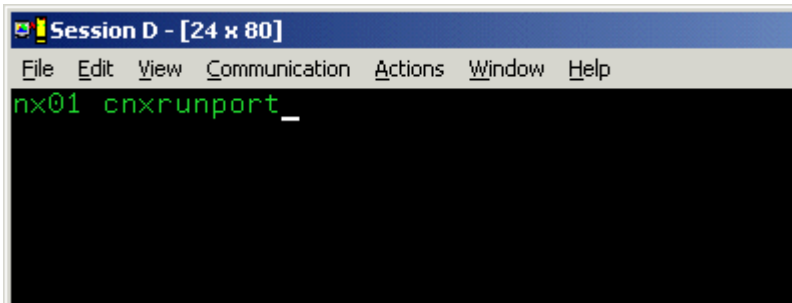


2. The variable is removed, and verification appears.

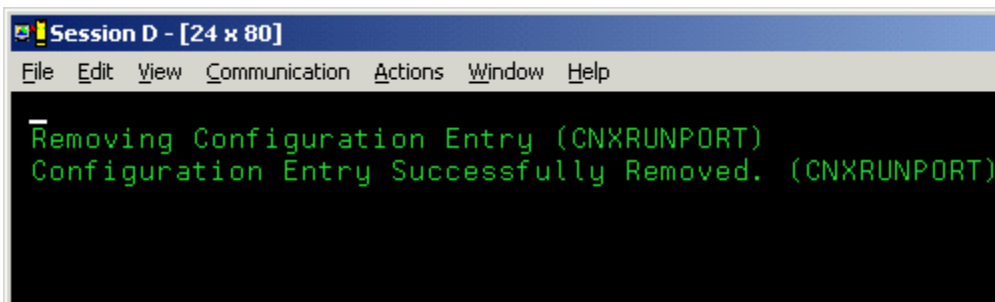


To convert the CONNX port number to the default

1. Clear the screen, type **NX01 CNXRUNPORT**, and then press **<Enter>**.

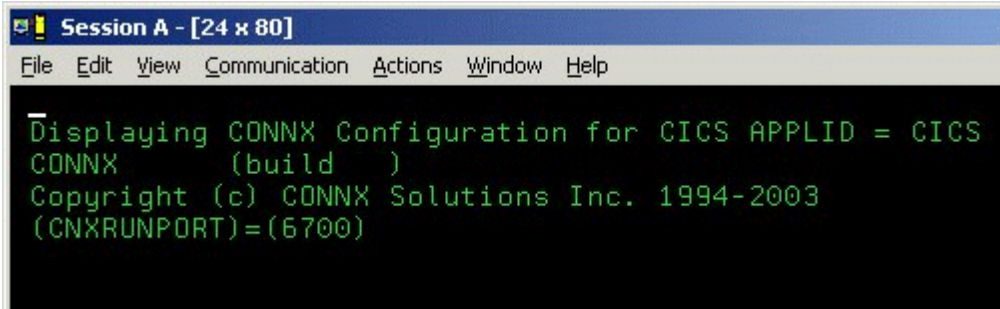


2. The current entry for CNXRUNPORT is removed, and verification appears. Removal of the CNXRUNPORT variable restores the default value of 6500.



To display the current CONNX environment variables

1. Type **NX01**, and then press the **<Enter>** key to display the current configuration.
2. To test your connection, log on to CONNX and follow the import procedures described in CONNX and VSAM in the CONNX User Reference Guide available online, on the CONNX CD-ROM, and within the CONNX product.



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
_
Displaying CONNX Configuration for CICS APPLID = CICS
CONNX      (build    )
Copyright (c) CONNX Solutions Inc. 1994-2003
(CNXRUNPORT)=(6700)
```

Part 4: Uninstallation Procedures for CONNX for CICS/VSAM - z/OS**Running CONNX for CICS/VSAM - z/OS Uninstallation Without Shutting Down CICS**

You can perform a partial uninstallation of CONNX without shutting down the target CICS region as follows:

1. Log on to the target CICS region and stop the CONNX CICS TCP/IP Listener transaction NX00 by entering:

NX01 STOP

2. Next, use the CEMT transaction to close and disable the CONNX VSAM configuration and sample files:

CEMT SET FI(CNX*) CLO DIS

3. Edit the @REMOVE JCL member and delete all references to the CONNX load library and the CONNX SAS/C QSAM sequential data sets:

CONNX.CICS.STDERR
CONNX.CICS.STDOUT
CONNX.CICS.STGRPT

4. Submit the modified JCL and navigate to a different ISPF option so that the CONNX.VVRR.CNTL PDS is also deleted:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
File Edit Confirm Menu Utilities Compilers Test Help

EDIT      CONNX.CICS.CNTL(@REMOVE) - 01.00          Columns 00001 00072
Command ==> save;sub;=6                          Scroll ==> CSR
***** ***** Top of Data *****
000001 //CNXREMOV  JOB (ACCT#),
000002 //          'CONNX      UNINSTALL',
000003 //          MSGCLASS=H,
000004 //          MSGLEVEL=(1,1),
000005 //          NOTIFY=CONNX ,
000006 //*         USER=,
000007 //*         PASSWORD=,
000008 //*         REGION=2M,
000009 //*         TIME=(2,59),
000010 //*
000011 //*----- CONNX for VSAM Version -----*
000012 //*----- Copyright 2003 by CONNX Solutions Inc. -----*
000013 //*
000014 //* MEMBER:  @REMOVE0
000015 //* PURPOSE: THIS JOB REMOVES ALL CONNX COMPONENTS IN
000016 //*           TWO STEPS.
000017 //*
000018 //* =====
000019 //* =====
M1 a 04/015

```

5. After this job completes and the target CICS region is shut down, you can follow the steps described in Manual Steps: CONNX for CICS/VSAM - OS/390 / z/OS to complete the CONNX uninstallation.

VSE

CONNX for VSAM - VSE

Prerequisites - CONNX for VSAM-VSE

The IBM VSE target system must be configured to support TCP/IP communications software from CSI international, IBM or Barnard Software, Inc..

For VSE installations, the only other setup step required is to concatenate the VSE TCP/IP library to the target CICS partition LIBDEF search list. For example, given the default VSE TCP/IP, CONNX and IBM Language Environment library names:

```
// LIBDEF *,SEARCH=(CNXVRR.RUNLIB,           X
      PRD2.TCPIP,                             X
      PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD, X
      PRD2.SCEECICD,PRD2.SCEECICS,PRD2.SCEEBASD, X
      PRD2.SCEEBASE,PRD2.DBASE),PERM
```

Verifying TCP/IP Connectivity

Before you install CONNX on your VSAM server, we recommend that you verify the TCP/IP connectivity to your VSE system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

```
ping hostname
```

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Part 1: Creating the CONNX Library in VSAM-managed Space

Creating the CONNX Library

CONNX for VSAM/VSE provides an automated installation client application that transfers the CONNX server components to the target VSE host via TCP/IP FTP. To succeed, the automated FTP step requires a target VSE library defined in VSAM-managed space. The target library must be defined manually, either via the following CONNX-supplied template and FTP, or by using the VSE resource-definition panels. To use the CONNX installation template, follow the steps below:

1. The CONNX VSE installation template is copied to the CONNX VSE installation directory on the client machine; the default name is

C:\connx32\vsam\templates\ftp\vse\cnxlibv\$.dat

2. Using your editor of choice, such as MS Notepad, Wordpad, Write, etc., open the cnxlibv\$.dat file and make the following substitutions for the template variables defined in the following table:

CONNX for VSAM - VSE Installation Template Substitutable Variables

Variable	Definition
[CNXLIB]	The 1-7 character CONNX VSAM library name defined in the [CNXLBP] CONNX library VSAM cluster; the default is CNXVRR.
[CNXLBP]	The 1-44 character fully-qualified VSE VSAM cluster name which contains the CONNX V08R08 VSAM library; the default is CONNX..LIBRARY.
[CNXCATL]	The 1-7 character VSE standard label which references the physical data set name of the VSAM user or system catalog ([CNXCATP]).
[CNXCATP]	The 1-to-44 character physical data set name of the VSE VSAM user catalog into which the CONNX VSAM-managed library [CNXLBP] is defined. If left blank, the CONNX library is defined in the default VSE system catalog.
[SAMPLEVOLSER]	The 1-6 character DASD volume serial number on which the CONNX VSE VSAM-managed library is installed: no default.

A sample edit session appears below:

```

cnxlibv$ - Notepad
File Edit Format Help
* $$$ JOB JNM=CNXLIB,CLASS=0,DISP=D,NTFY=YES
// JOB CNXLIB DEFINE CONNX for VSAM/VSE VSAM-managed library
// EXEC LIBR,PARM='MSHP'
//      DELETE SUBLIB=[CNXLIB].CNTL
//      DELETE SUBLIB=[CNXLIB].COPYBOOK
//      DELETE SUBLIB=[CNXLIB].OBJLIB
//      DELETE SUBLIB=[CNXLIB].RUNLIB
//      DELETE SUBLIB=[CNXLIB].SOURCE
//
// EXEC IDCAMS,SIZE=AUTO
//      DELETE [CNXLBP]          -
//      CLUSTER PURGE           -
//      CATALOG ([CNXCATP])
//
// OPTION STDLABEL=DELETE
//      1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
//      [CNXLIB]
//
// EXEC IESVCLUP,SIZE=AUTO
//      1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
//      [CNXLIB]
//

```

Replace [?] [X]

Find what: [CNXLIB] [Find Next]

Replace with: CNXVRR [Replace]

[Replace All]

[Cancel]

☐ Match case

Replace [?] [X]

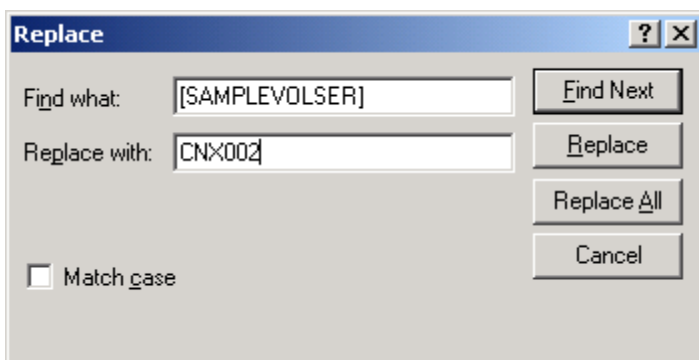
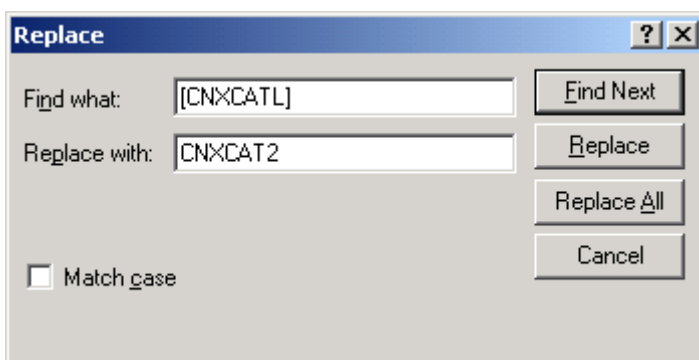
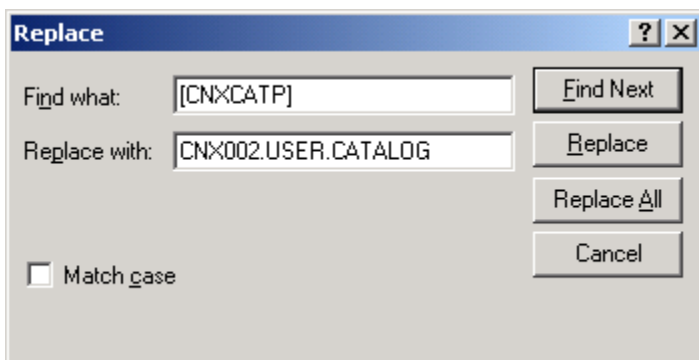
Find what: [CNXLBP] [Find Next]

Replace with: CONNX.LIBRARY [Replace]

[Replace All]

[Cancel]

☐ Match case



3. After making the substitutions in the cnxlibv\$.dat template, visually inspect the JCL and move each line continuation character (a rightmost capital X preceded by a comma and one or more blanks) to column 72; truncate all data beyond column 72.

The control records for the standard label delete and add job steps are column-sensitive; examples of unsubstituted and substituted control records follow:

Column Rule Lines

Delete Step, template version:


```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// DLBL [CNXLIB], '[CNXLIB]', ,VSAM, CAT=[CNXCATL], DISP=(OLD, KEEP)
/*

```

Substituted example:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// DLBL CNXV8R9, 'CONNX.LIBRARY', ,VSAM, CAT=CNXCAT2, DISP=(OLD, KEEP)
/*

```

Add Step, template version:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// EXEC IESVCLUP, SIZE=AUTO
A [CNXLIB]
/*

```

Substituted example:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// EXEC IESVCLUP, SIZE=AUTO
A CONNX.LIBRARY
/*

```

4. Delete the column rule lines in the template and save the cnxlibv\$.dat file as a new file; for example, cnxlibv0.jcl.
5. Via the Windows command prompt, start an ftp session to the target VSE host. Change directory to the VSE POWER Reader queue and submit the cnxlibv0.jcl job:


```

>ftp
>open 123.456.788.812
>user [USER]
>password [PASSWORD]
>cd POWER\RDR\A
>put cnxlibv0.jcl

```
6. This job should execute in less than a minute. Within the same ftp session, retrieve the output JCL from the POWER List Queue:


```

>cd "..\..\LST\A"
>dir

```
7. Given the default job name defined in the cnxlibv\$.dat template, a job with the following naming convention is displayed:

- CNXLIB.00037.00 04/17/03 08:32 94 11 79 3 D SYSTCPIP
8. Retrieve the job output by issuing the following command:
 >get CNXLIB.00037.00
9. The sample output below is from a run which deletes CONNX sub-libraries and libraries in VSAM-managed space before redefining new versions:

```

1// JOB  CNXLIB DEFINE CONNX FOR VSAM/VSE VSAM-MANAGED LIBRARY
      DATE 04/17/2003, CLOCK 08/32/01
      // EXEC LIBR,PARM='MSHP'
1      DELETE SUBLIB=CNXVVR.CNTL
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVVR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVVR.COPYBOOK
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVVR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVVR.OBJLIB
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVVR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVVR.RUNLIB
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVVR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVVR.SOURCE
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVVR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
11S55I LAST RETURN CODE WAS 0008
      // EXEC IDCAMS,SIZE=AUTO
1IDCAMS SYSTEM SERVICES                                TIME: 08:32:02
      04/17/2003   PAGE   1
0
      DELETE CONNX..LIBRARY                          -
      CLUSTER PURGE                                  -
      CATALOG (CNX002.USER.CATALOG)
0IDC3012I ENTRY CONNX.LIBRARY NOT FOUND
      IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLCG-6
      IDC0551I **ENTRY CONNX.LIBRARY NOT DELETED
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 8
11S55I LAST RETURN CODE WAS 0008
      // OPTION STDLABEL=DELETE
      CNXVVR
1L63I LABEL CNXVVR NOT FOUND
      // EXEC IESVCLUP,SIZE=AUTO
1// EXEC IDCAMS,SIZE=AUTO
  
```

```

1IDCAMS  SYSTEM SERVICES                                TIME: 08:32:05
      04/17/2003  PAGE  1
0
  DEFINE CLUSTER ( -
      NAME ( CONNX..LIBRARY )                          -
      TRACKS (300 30 )                                  -
      SHAREOPTIONS (3)                                  -
      RECORDFORMAT (NOCIFORMAT)                         -
      VOLUMES (CNX002)                                  -
      NOREUSE                                             -
      NONINDEXED                                         -
      TO (99366))                                        -
      DATA (NAME ( CONNX..LIBRARY.@D@ ) )              -
      CATALOG (CNX002.USER.CATALOG)
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
      IF LASTCC NE 0 THEN CANCEL JOB
0IDC0204I PRECEDING COMMAND BYPASSED DUE TO CONDITION CODES
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
11S55I  LAST RETURN CODE WAS 0000
      // OPTION STDLABEL=ADD
      // DLBL CNXVVR, 'CONNX..LIBRARY',,VSAM,CAT=CNXCAT2,DISP=(OLD,KEEP)
      // EXEC IESVCLUP,SIZE=AUTO
1// EXEC LIBR,PARM='MSHP'
1      DEFINE LIB=CNXVVR      REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.CNTL      REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.COPYBOOK REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.OBJLIB    REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.RUNLIB    REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.SOURCE    REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
11S55I  LAST RETURN CODE WAS 0000
      EOJ CNXLIB      MAX.RETURN CODE=0008      DATE
      04/17/2003, CLOCK 08/32/13,      DURATION 00/00/12

```

Notes:

For a first-time run, the return codes from the first step:

// EXEC LIBR,PARM='MSHP'

will be 8, which is expected.

10. For a first-time run, the return code from the IDCAMS statement:

DELETE CONNX.VXXRXX.LIBRARY -

CLUSTER PURGE -

CATALOG (CNX001.USER.CATALOG)

will be 8, which is expected. All subsequent return codes should be 0 or 4.

11. The VSE volume and VSAM user catalog physical data set and data label names in the above example are CNX002, CNX002.USER.CATALOG, and CNXCAT1. These values are substituted into template variables [SAMPLEVOLSER], [CNXCATP], and [CNXCATL].
12. Make the CONNX for VSAM - VSE library available for FTP access.
13. Depending on the TCP/IP security rules of the target VSE system, the newly allocated CONNX library may not be immediately available for ftp read/write access. The CONNX VSAM library can be manually defined to the VSE TCP/IP partition via VSE console commands as follows:

msg f7

AR 0015 1140I READY

F7-0100 IPN300I Enter TCP/IP Command

102 def file,pub=CNXVVRR,dlbl=CNXVVRR,type=lib,READONLY=NO

F7 0098 IPN264I File defined, Dataset: CNXVVRR

The CONNX for VSE-VSAM library can be permanently defined to the TCP/IP partition via the batch JCL provided in member IPINIT00 of the TCP/IP for VSE installation library. The control record syntax is identical to the console command:

DEFINE FILE,PUBLIC='CNXVVRR',DLBL=CNXV8R8,TYPE=LIBRARY,READONLY=NO

To verify that the new CNXVVRR library is defined and available for ftp, start a Command Prompt window, and then an ftp session to the target host. Once logged on, you can navigate to the CNXVVRR directory and list the subdirectories:

230 User logged in, proceed.

ftp> cd cnxv8r9

250 Requested file action okay, completed.

ftp> dir

200 Command okay.

150 File status okay; about to open data connection

CNTL <Sub Library> 0 1 04/17/03 08:32

COPYBOOK <Sub Library> 0 1 04/17/03 08:32

OBJLIB <Sub Library> 0 1 04/17/03 08:32

RUNLIB <Sub Library> 0 1 04/17/03 08:32

SOURCE <Sub Library> 0 1 04/17/03 08:32

14. You are ready to begin the CONNX for VSAM/VSE automated installation. Click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX Server Setup**. Follow the steps described in the next section, "Part 2: Installing the CONNX VSAM Components on a VSE Server".

Part 2: Installing the CONNX VSAM Components on a VSE System

General Information about the CONNX VSAM server

This step copies the CONNX VSAM server components from a client PC to a VSE target. The CONNX Administrator component must be installed on the client PC.

About the VSAM-VSE User Account and Password

When installing the VSAM components, select a VSE user account name and password with privileges that allow you to create VSE libraries in VSAM-managed space and to copy, read, and write files via FTP to the target host.

To install the CONNX for VSAM-VSE components

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the **VSE** tab in the CONNX Server Component Installation dialog box.

The screenshot shows the 'CONNX Server Component Installation' dialog box with the 'VSE' tab selected. The dialog box has a title bar with the text 'CONNX Server Component Installation' and standard window controls. Below the title bar is a 'Files' section. The main area is titled 'CONNX Server Component Installation' and contains the instruction 'Enter the platform type, transfer method and login credentials.' Below this is a tabbed interface with 'z/OS', 'VSE', 'VMS', and 'UNIX/Linux' tabs. The 'VSE' tab is active. The dialog box is divided into several sections: 'CONNX Server Library' with fields for 'CONNX Library DSN' (containing 'CONNX'), 'VOLSER', 'CONNX Library Label' (containing 'CNXVXRX'), 'Catalog DSN', and 'Catalog Label'; 'Login Information' with fields for 'Server', 'Username', and 'Password'; 'Product' with radio buttons for 'ADABAS' and 'VSAM' (selected); 'TCP/IP Stack' with radio buttons for 'CSI 1.5E and below', 'CSI 1.5E and above' (selected), and 'Barnard'; 'Mode' with radio buttons for 'Active' and 'Passive' (selected); 'Load Library Parameters' with a field for 'Target RDO File' (containing 'CICS.CSD'); and 'VSAM Samples' with a 'Create' checkbox (checked), 'VOLSER' field, and 'Database HLQ' field (containing 'CONNX'). At the bottom right are 'Install' and 'Done' buttons.

2. In the **CONNX Server Library** area, enter the fully qualified VSE cluster name which contains the CONNX VVRR library in **CONNX Library DSN** and its VOLSER in the **VOLSER** text box. Enter the CONNX library name defined in the CONNX library cluster in **CONNX Library Label**, the data set name of the VSE user catalog where the CONNX sample files and managed library are defined in **Catalog DSN**, and the system data label which references the physical data set name of the User Catalog in **Catalog Label**.

Note: If **CONNX Library Label** is blank, the library name defaults to CNXWRR. If **Catalog DSN** is blank, CONNX files are defined in the default VSE system catalog. If **Catalog Label** is blank, the data label defaults to IJSYSCT.

3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your VSE server in the **Server** text box, a privileged user account name in the **User ID** text box and its password in the **Password** text boxes. Under **Product**, click **VSAM** and under **Server Type**, click **CICS**.

4. In the **TCP/IP Stack** area, if you are using the Barnard TCP/IP stack rather than CSI, select the **Barnard** radio button. For **Mode**, select either **Active** or **Passive** FTP mode. Unless your system is not configured to support Passive mode FTP, you should select Passive. If the FTP transfer fails for any reason other than the user id or password was entered incorrectly, switch modes and try again.

Enter the TCP/IP stack location in **TCP/IP HLQ**.

Caution: To enable the environment variable for the Barnard TCP/IP stack, set [CNXBARNARD](#) to 1 in the CONNX Configuration Utility.

5. In the **CICS Parameters** area, see the table in [Required and Optional Entry Fields for VSAM TCP/IP Server Setup](#) for a detailed explanation of the information required in **Target RDO File**.
6. In the **VSAM Samples Group** area, if you clear the **Create** check box, CONNX will not create the CONNX VSAM sample KSDS, RRDS, and ESDS data sets. If the **Create** check box is selected, the CONNX VSAM sample KSDS, RRDS, and ESDS data sets and the VSAM clusters described in [VSAM Objects for CONNX.CICS.SAMPLES](#) are created, defined to the target CICS region, and given the data set prefix specified in **Database HLQ**.
Note: The default dataset prefix for **Database HLQ** is CONNX.SAMPLES.
7. Click the **Install** button. This action starts an FTP session and copies the CONNX components from the client PC to the target host. A message confirming that the FTP session has started appears in the status bar..
8. Allow several minutes for the file transfer process to complete.
9. Click the **Close** button. The **CONNX Server Component Installation** dialog box closes.

Required and Optional Entry Fields for VSAM TCP/IP Server Setup

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS **VSE** **VMS** **UNIX/Linux**

CONNX Server Library

CONN~~X~~ Library DSN: CONNX

VOLSER:

CONNX Library Label: CNXVXRX

Catalog DSN:

Catalog Label:

Login Information

Server:

Username:

Password:

Load Library Parameters

Target RDO File: CICS.CSD

VSAM Samples

Create ☒ VOLSER:

Database HLQ: CONNX

Product

☐ ADABAS

☒ VSAM

TCP/IP Stack

☐ CSI 1.5E and below

☒ CSI 1.5E and above

☐ Barnard

Mode

☐ Active

☒ Passive

PRD2.TCPIP

Install **Done**

Group Box	Term	Description
Login Information	Server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.123
	User ID	Your 1-8 character VSE logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character VSE password. All characters convert from lowercase to uppercase.
TCP/IP Stack	CSI	CSI International TCP/IP connectivity.
	Barnard	The Barnard TCP/IP communications stack.
	TCP/IP HLQ	The TCP/IP symbolic host name or dotted numeric address for your VSE system.
CONNX Server Library	CONNX Library DSN	A one- or multi-part high-level data set name, which is used to create the CONNX installation sequential and partitioned data sets on the target host. In the current example, the library DSN is

		CONNX.CICS.
	VOLSER	The DASD volume serial on which the CONNX.Adabas sequential and partitioned data sets are created. Optional: The DASD volume serial on which the CONNX.ADABAS sequential and partitioned data sets are created.
	CONNX Library Label	The 1-44 character fully qualified VSE Adabas SQL Cluster name that contains the CONNX VVRR Adabas SQL Gateway library. The default is CNXV9R0.
	Catalog DSN	The 1-44 character physical data set name of the VSE Adabas SQL Gateway user catalog into which the Adabas SQL Gateway-managed library are defined. If left blank, the CONNX files are defined in the default VSE system catalog.
	Catalog Label	The 1-7 character system data label which references the physical data set name of the Adabas SQL Gateway User Catalog.
CICS Parameters	Target RDO File	System Management File location.
VSAM Samples	Database HLQ	A one- or multi-part high-level data set prefix, which is used to create the CONNX VSAM sample KSDS, RRDS, and ESDS data sets.
	Create	Creates VSAM sample KSDS, RRDS, and ESDS data sets if checked. If cleared, no sample files are created.
	VOLSER	The DASD volume serial on which the CONNX.Adabas sample data sets are created.

VSAM Objects for CONNX.CICS.SAMPLES

VSAM Object	Description	CICS File Name	CONNX for VSAM Sample Table Name
CONNX.CICS.SAMPLES.CUSTOMER	Customer file KSDS.	CNXCSTK	CUSTOMERS_VSAM
CONNX.CICS.SAMPLES.CUSTOMER.E	Customer file ESDS.	CNXCSTE	CUSTOMERS_VSAM_ESDS
CONNX.CICS.SAMPLES.CUSTOMER.R	Customer file RRDS.	CNXCSTR	CUSTOMERS_VSAM_RRDS
CONNX.CICS.SAMPLES.EQPATH01	Equipment file path #1.	CNXEQP1	*none*
CONNX.CICS.SAMPLES.EQPATH01	Equipment file path #1.	CNXEQP2	*none*
CONNX.CICS.SAMPLES.EQUIPMNT	Equipment file KSDS.	CNXEQK	EQUIPMNT_VSAM
CONNX.CICS.SAMPLES.EQUIPMNT.E	Equipment file ESDS.	CNXEQE	EQUIPMNT_VSAM_ESDS
CONNX.CICS.SAMPLES.EQUIPMNT.R	Equipment file RRDS.	CNXEQR	EQUIPMNT_VSAM_RRDS
CONNX.CICS.SAMPLES.ORDER	Order file KSDS.	CNXORK	ORDERS_VSAM
CONNX.CICS.SAMPLES.ORDER.E	Order file ESDS.	CNXORE	ORDERS_VSAM_ESDS
CONNX.CICS.SAMPLES.ORDER.R	Order file RRDS.	CNXORR	ORDERS_VSAM_RRDS
CONNX.CICS.SAMPLES.ORDPTH01	Order file path #1.	CNXORP1	*none*
CONNX.CICS.SAMPLES.ORDPTH02	Order file path #2.	CNXORP2	*none*
CONNX.CICS.SAMPLES.PRDGRP	Product group file KSDS.	CNXPBK	PRODUCT_GROUPS_VSAM
CONNX.CICS.SAMPLES.PRODGRP.E	Product Group file ESDS.	CNXPGE	PRODUCT_GROUPS_VSAM_ESDS
CONNX.CICS.SAMPLES.PRODGRP.R	Product file RRDS.	CNXPGR	PRODUCTS_GROUPS_VSAM_RRDS
CONNX.CICS.SAMPLES.PRODUCT	Product file KSDS.	CNXPRDK	PRODUCTS_VSAM
CONNX.CICS.SAMPLES.PRODUCT.E	Product	CNXPRD	PRODUCTS_VSAM_ESDS

Step 5 - Install IBM Mainframe-Compatible Server

	file ESDS	E	
CONNX.CICS.SAMPLES.PRODUCT.R	Product file RRDS.	CNXPRD R	PRODUCTS_VSAM_RRDS

Part 3: Installing the CONNX CICS components

To install the selected components to the VSE target

For CICS/VSE 2.3 and above installations:

- The initial FTP step creates an installation JCL member on the target VSE system in the CONNX VSAM library. Given the default physical VSAM file name of CONNX.VVRR.LIBRARY and the default library.sublib name of CNXVVRR.CNTL, submit the JCL by starting an interactive DITTO session, editing member CNXINST0.JCL to conform to your installation standards, and submitting the revised JCL to the VSE reader queue.
- Save the changes to CNXINST0.JCL with the DITTO Save command, then submit the JCL to the VSE POWER reader queue with the DITTO submit command.

For CICS/VSE 2.3 and below installations:

- The initial FTP step creates an installation JCL member on the target VSE system in the CONNX VSAM library. Given the default physical VSAM file name of CONNX.LIBRARY and the default library.sublib name of CNXVVRR.CNTL, submit the JCL by starting an interactive DITTO session, editing member CNXINS10.JCL to conform to your installation standards, and submitting the revised JCL to the VSE reader queue.

Summary of the CONNX for VSAM-VSE Installation Procedure

For CICS/VSE 2.4 and above installations:

Steps 1-4: Execute the linkage editor against object code for the following CONNX programs:

CNXCFCG	Configuration + TCP/IP Listener start/stop program
CNXRUN	TCP/IP Listener
CNXVSD0	TCP/IP Server
CNXSTOP	Automated shutdown program for CNXRUN

The four link-edit steps must end in return codes <= 4.

Step 5: Executes the IBM batch CICS RDO update utility program DFHCSDUP to define CONNX files, transactions, and programs to the target CICS partition RDO VSAM cluster. The first execution returns warnings for the unconditional DELETES of CONNX objects from the undefined RDO Group CNXVVR. Such warnings can be safely ignored. After all CONNX RDO objects are defined, a list of the objects created in RDO group is generated.

Step 6: Executes the IBM Access Methods Services utility program IDCAMS to DELETE/DEFINE the CONNX VSAM configuration file (CNXCFCG) and the CONNX VSAM sample KSDS and RRDS files.

Step 7: Executes the IBM Access Methods Services utility program IDCAMS to copy (REPRO) the CONNX VSAM sample data from the temporary ESDS files created by the initial FTP step to the corresponding KSDS and RRDS files. After the REPRO steps, the temporary ESDS files are deleted, reallocated, and loaded with data from the corresponding KSDS files. This step saves VSAM space. Finally, two VSAM alternate indexes each for the CONNX VSAM sample KSDS equipment and order files are defined and built. The return code from this step should be <= 4.

Step 8: To submit the CONNX for VSE batch installation job to the VSE POWER reader queue:

1. Change the job name, class, and PDEST/LDEST keywords to match installation standards.
2. For this example, all CONNX for VSE modules are defined in the CNXVVR library which resides in VSAM dataset CONNX.LIBRARY in VSAM user catalog CNXCAT2.
3. The // LIBDEF OBJ search list points to the default lib.sublib for the CONNX for VSE/VSAM object code (CNXVVR.OBJLIB); the PRD2.TCPIP library is the default target for the TCP/IP for VSE modules from Connectivity Systems, Inc. Change these entries as necessary to match your VSE installation standards.

Save the changes to CNXINST0.JCL; then enter submit on the DITTO command line to copy the JCL to the VSE POWER reader queue.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CNXINST0.JCL      Library CNXVVR.R.CNTL      Col 1      Format CHAR
                                     SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 * $$ JOB JNM=CNXINST,PDEST=(*,CNX),LDEST=(*,CNX),CLASS=0
00002 // JOB      CNXINST
00003 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM,                      X
00004           CAT=CNXCAT2,DISP=(OLD,KEEP)
00005 // LIBDEF  OBJ,SEARCH=(CNXVVR.R.OBJLIB,PRD2.TCPIP)
00006 // LIBDEF  PHASE,CATALOG=CNXVVR.R.RUNLIB
00007 // OPTION  CATAL
00008 ACTION    MAP
00009 PHASE     CNXCFG,*
00010 INCLUDE   DFHEAI
00011 INCLUDE   DFHEAIO
00012 INCLUDE   CNXCFG

Command ==> submit      Scroll CSR
F1=Help  F2=SplitJoin  F3=Save+Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd
F8=Fwd   F10=RChange  F11=CRetrieve  F12=Quit

MA a 22/021

```

For CICS/VSE 2.3 and below installations:

Steps 1-4: Execute the linkage editor against object code for the following CONNX programs:

CNXCFCG	Configuration + TCP/IP Listener start/stop program
CNXRUN	TCP/IP Listener
CNXVSD0	TCP/IP Server
CNXSIGN	CONNX CICS/VSE 2.3 External Security Interface program

The four link-edit steps must end in return codes <= 4.

Step 5: Executes the IBM batch CICS RDO update utility program DFHCSDUP to define CONNX transactions and programs to the target CICS partition RDO VSAM cluster. The first execution returns warnings for the unconditional DELETES of CONNX objects from the target RDO group(CNXVVR.R) which is not defined. Such warnings can be safely ignored. After all CONNX RDO objects are defined, a list of the new RDO group is generated.

Step 6: Executes the IBM Access Methods Services utility program IDCAMS to DELETE/DEFINE the CONNX VSAM configuration file (CNXCFG) and the CONNX VSAM sample KSDS and RRDS files.

To submit the CONNX for VSE batch installation job to the VSE POWER reader queue:

1. Change the job name, class, and PDEST/LDEST keywords to match installation standards.
2. For this example, all CONNX for VSE modules are defined in the CNXVVR.R library which resides in VSAM dataset CONNX.LIBRARY in VSAM user catalog CNXCAT1.

3. The // LIBDEF OBJ search list points to the default lib.sublib for the CONNX for VSE/VSAM object code (CNXVVR. OBJLIB); the PRD2.TCPIP library is the default target for the TCP/IP for VSE modules from Connectivity Systems, Inc. Change these entries as necessary to match your VSE installation standards.

```
// LIBDEF OBJ,SEARCH=(CNXVVR.OBJLIB,PRD2.TCPIP,PRD2.CICSOLDP)
```

PRD2.CICSOLDP is the default library for the CICS/VSE 2.3 Coexistence Environment.

Step 7: Executes the IBM Access Methods Services utility program IDCAMS to copy (REPRO) the CONNX VSAM sample data from the temporary ESDS files created by the initial FTP step to the corresponding KSDS and RRDS files. After the REPRO steps, the temporary ESDS files are deleted, reallocated, and loaded with data from the corresponding KSDS files. This step saves VSAM space.

Finally, two VSAM alternate indexes each for the CONNX VSAM sample KSDS equipment and order files are defined and built. The return code from this step should be <= 4.

Save the changes to CNXINS10.JCL; then enter submit on the DITTO command line to copy the JCL to the VSE POWER reader queue.

ICS/VSE 2.3 External Security Interface: Prerequisites and Setup

CONNX for CICS/VSE 2.3 and below verifies CICS userid/password combinations via the program CNXSIGN, which invokes the CICS/VSE 2.3 External Security Program DFHXSP. The default DFHXSP program requires a terminal ID during userid/password verification; first it loads the CICS sign-on table (DFHSNT); next, it optionally invokes external security manager program DFXSE. Both default programs can be replaced by user-written or third-party vendor-supplied versions, which must conform to standard call/return parameter blocks. For more information on setup of the CICS/VSE 2.3 External Security Interface, refer to the following document:

Title: *CICS/VSE V2R3 Customization Guide*

Document Number: SC33-0707-02

For CICS 2.3 Coexistence Environments, the following steps are necessary to install the External Security Interface:

1. Rename the original program phases for DFHXSE and DFHXSSCO, and reassemble the versions in library.sublib PRD2.CICSOLDP;
2. Define program DFHXSSCO to CICS/VSE via the CEDA command:

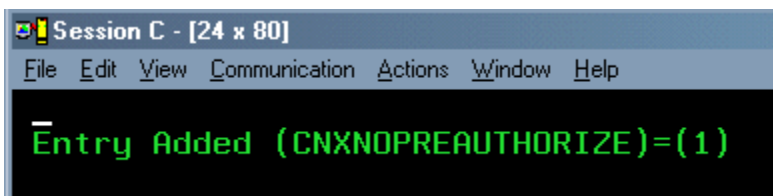
```
CEDA DEF PROG(DFHXSSCO) GROUP(VSESPO) LANG(AS) RES(YES)
RSL(PUBLIC)
```
3. Define a default DFHSNT (CICS Signon Table) entry;
4. Code EXTSEC=YES in the CICS System Initialization Table (DFHSIT) source and reassemble;
5. Restart CICS/VSE.

Bypassing the CONNX CICS/VSE 2.3 External Security Interface

By default, the CONNX CICS/VSE 2.3 userid/password verification logic invokes the External Interface program DFHXSP. This logic can be bypassed by setting CONNX environment variable CNXNOPREAUTHORIZE to 1:



```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
nx01 cnxnopreauthorize 1_
```



```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
Entry Added (CNXNOPREAUTHORIZE)=(1)
```

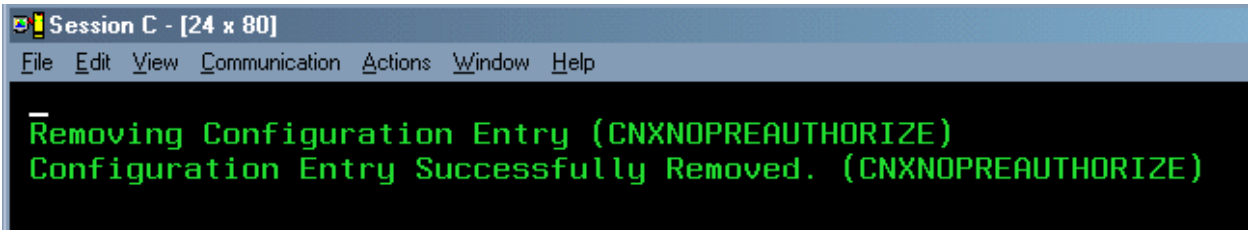
Caution: When CNXNOPREAUTHORIZE is set to 1, userid/password verification is bypassed. This means that the CONNX TCP/IP Listener and Server programs inherit the security attributes of the CICS userid which starts the CONNX TCP/IP Listener transaction (NX00) via the NX01 START command. This setting can be used during initial installation and testing, but it is strongly recommended that userid / password checking be enabled for production installations.

Enabling the CONNX CICS/VSE 2.3 External Security Interface

To enable userid/password verification, delete the CNXNOPREAUTHORIZE environment variable via the NX01 transaction:



```
Session C - [24 x 80]  
File Edit View Communication Actions Window Help  
nx01 cnxnopreauthorize_
```



```
Session C - [24 x 80]  
File Edit View Communication Actions Window Help  
-  
Removing Configuration Entry (CNXNOPREAUTHORIZE)  
Configuration Entry Successfully Removed. (CNXNOPREAUTHORIZE)
```

Completing CICS Customization

Once the CONNX for VSAM - VSE CICS components are built, they must be defined to the target CICS partition. This process includes updating the CICS DCT (Destination Control Table), and adding the CONNX library to the CICS LIBDEF SEARCH list.

For **CICS/VSE 2.3** installations, the CICS File Control Table (FCT) needs to be updated with the CONNX VSAM Configuration and sample file definitions, and then reassembled.

To define the SAS/C 7.00 CICS extrapartition data queues in the CICS DCT

The CONNX for VSAM - VSE CICS programs require three Destination Control Table changes to define three extra-partition transient data queues (TDQs) for tracing and debugging. The example DCT source below defines SAS/C TDQs SASO, SASE, and SASR.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LB - Library Member Browse

Member SASDCT.ASM          Library CNXVVRR.SOURCE      Col 1          Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 STDOUT DFHDCT TYPE=SDSCI,          USED BY SAS/C FOR STDOUT      +
00002          DSCNAME=STDOUT,          +
00003          BLKSIZE=137,              +
00004          RECSIZE=133,              +
00005          RECFORM=VARUNB,           +
00006          TYPEFLE=OUTPUT,           +
00007          BUFNO=1                   +
00008 STDERR DFHDCT TYPE=SDSCI,          USED BY SAS/C FOR STDERR      +
00009          DSCNAME=STDERR,           +
00010          BLKSIZE=1370,              +
00011          RECSIZE=133,              +
00012          RECFORM=VARBLK,           +

Command ==>                      Scroll CSR
F1=Help F2=Zoom F3=Exit F4=Left F5=Right F6=RFind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

MA a 22/015

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LB - Library Member Browse

Member SASDCT.ASM          Library CNXVVAR.SOURCE      Col 1          Format CHAR
                               SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00013          TYPEFLE=OUTPUT,                          +
00014          BUFNO=1                                     +
00015 *
00016 *          RECSIZE IS FOR WIDE REPORTS, PREFIXED WITH TERMID/TRANID,
00017 *          PLUS FOUR BYTES FOR RECORD LENGTH (LLBB)
00018 *
00019 STGRPT  DFHDCT TYPE=SDSCI,          USED BY SAS/C FOR =STORAGE REPORTS +
00020          DSCNAME=STGRPT,              +
00021          BLKSIZE=1450,                  +
00022          RECSIZE=145,                  133 + 8 + 4      +
00023          RECFORM=VARBLK,              +
00024          TYPEFLE=OUTPUT,              +
00025          BUFNO=1

Command ==>          Scroll CSR
F1=Help  F2=Zoom  F3=Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd  F8=Fwd
F10=Actions  F11=CRetrieve  F12=Cancel

MA          a          22/015

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LB - Library Member Browse

Member SASDCT.ASM          Library CNXVVAR.SOURCE      Col 1          Format CHAR
                               SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00026 SASO    DFHDCT TYPE=EXTRA,                          +
00027          DESTID=SASO,          USED BY SAS/C FOR STDOUT      +
00028          DSCNAME=STDOUT
00029          RSL=PUBLIC          ALLOW ANY TRANSACTION TO ACCESS
00030 SASE    DFHDCT TYPE=EXTRA,                          +
00031          DESTID=SASE,          USED BY SAS/C FOR STDERR      +
00032          DSCNAME=STDERR
00033          RSL=PUBLIC          ALLOW ANY TRANSACTION TO ACCESS
00034 SASR    DFHDCT TYPE=EXTRA,                          +
00035          DESTID=SASR,          USED BY SAS/C FOR =STORAGE REPORTS +
00036          DSCNAME=STGRPT
00037          RSL=PUBLIC          ALLOW ANY TRANSACTION TO ACCESS
00038 ***** End of data *****

Command ==>          Scroll CSR
F1=Help  F2=Zoom  F3=Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd  F8=Fwd
F10=Actions  F11=CRetrieve  F12=Cancel

MA          a          22/015

```

To add SAS/C transient data queue definitions to the CICS Startup JCL

The CICS target partition startup JCL must be modified to specify the new DLBLs/datasets defined as SAS/C 7.00 Transient Data Queues, for example:

```
// ASSGN SYS012,DISK,VOL=volser,SHR
// DLBL STDOUT,'CONNX.CICS.STDOUT',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
// DLBL STDERR,'CONNX.CICS.STDERR',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
// DLBL STGRPT,'CONNX.CICS.STGRPT',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
```

Notes:

1. *Substitute a valid VSE DASD volume for volser.*
2. *Substitute a valid track start address and the number of tracks or blocks for each instance of aaa,bbb above. The recommended allocation for STDOUT is at least 15 tracks; the STDERR and STGRPT data sets can be initially allocated with 5 tracks each.*
3. *The data set prefix CONNX.CICS is suggested but not required; change it as appropriate to conform to your installation standards.*

Installation Verification of CONNX CICS Components

Option 1 - RDO: If you request to install the CONNX CICS components, the installation job executes the IBM DFHCSDUP utility to define the CONNX programs, transactions, and VSAM configuration and sample files to the target CICS partition RDO VSAM file. You can verify the installation of the CONNX CICS components by logging on to the target CICS partition and issuing the following command:

```
CEDA EXPAND GR(CNX)
```

For CICS 2.4 and above installations, the following components are defined:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
EXPAND GR(CNX0809)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
CNXCFC    FILE      CNXVRR     03.106    07.51.24
CNXCSTE    FILE      CNXVRR     03.106    07.51.35
CNXCSTK    FILE      CNXVRR     03.106    07.51.34
CNXCSTR    FILE      CNXVRR     03.106    07.51.35
CNXEQE     FILE      CNXVRR     03.106    07.51.51
CNXEQI1    FILE      CNXVRR     03.106    07.51.53
CNXEQI2    FILE      CNXVRR     03.106    07.51.54
CNXEQK     FILE      CNXVRR     03.106    07.51.50
CNXEQP1    FILE      CNXVRR     03.106    07.51.55
CNXEQP2    FILE      CNXVRR     03.106    07.51.56
CNXEQR     FILE      CNXVRR     03.106    07.51.52
CNXORE     FILE      CNXVRR     03.106    07.51.47
CNXORI1    FILE      CNXVRR     03.106    07.51.57
CNXORI2    FILE      CNXVRR     03.106    07.51.58
CNXORK     FILE      CNXVRR     03.106    07.51.45
CNXORP1    FILE      CNXVRR     03.106    07.51.59
+ CNXORP2  FILE      CNXVRR     03.106    07.52.01

SYSID=CIC2 APPLID=PRODCICS
RESULTS: 1 TO 17 OF 31      TIME: 16.44.09  DATE: 03.107
PF 1 HELP 3 END 4 TOP 5 BOT 6 CRR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
MA b 24/002

```

For CICS/VSE 2.3 installations, the following components are defined:


```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
EXPAND GR(CNX0809)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
+ CNXDORR  FILE      CNXVVR    03.106 07.51.48
  CNXPGE   FILE      CNXVVR    03.106 07.51.43
  CNXPGK   FILE      CNXVVR    03.106 07.51.41
  CNXPGR   FILE      CNXVVR    03.106 07.51.44
  CNXPRDE  FILE      CNXVVR    03.106 07.51.38
  CNXPRDK  FILE      CNXVVR    03.106 07.51.37
  CNXPRDR  FILE      CNXVVR    03.106 07.51.39
  CNXCFG   PROGRAM   CNXVVR    03.106 07.51.27
  CNXRUN   PROGRAM   CNXVVR    03.106 07.51.24
  CNXSTOP  PROGRAM   CNXVVR    03.106 07.51.29
  CNXVSAM  PROGRAM   CNXVVR    03.106 07.51.26
  NXS0     TRANSACTION CNXVVR    03.106 07.51.27
  NX00     TRANSACTION CNXVVR    03.106 07.51.25
  NX01     TRANSACTION CNXVVR    03.106 07.51.28

RESULTS: 18 TO 31 OF 31
SYSID=CIC2 APPLID=PRODCICS
TIME: 16.50.51 DATE: 03.107
PF 1 HELP 3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
MA b 24/002

```

As needed, you can ask your CICS Administrator to add the CNXLIST list to the target CICS partition startup list. CNXLIST contains the CNXVVR group.

Option 2 - Non-RDO:

If the target CICS partition does not use RDO, you can define the CONNX CICS files and programs to the target FCT and PPT. The CONNX FCT entries are copied to members CNXFCT0 and CNXPPT0 of the CNXV8R8.SOURCE installation library.sublib. These entries must be manually copied to the master source for the CICS FCT and PPT and reassembled.

For **CICS/VSE 2.3** installations, the FCT (File Control Table) for the target CICS partition must be updated with the CONNX FCT entries in member V2R3FCT0.ASM of the CNXVVR.SOURCE installation library.sublib. These entries must be manually copied to the master source for the CICS FCT and reassembled.

List of Installed Components on the CICS/VSE Partition

After proper installation, the following components appear in the CNXVVRR library in a VSAM-managed space.

CNXVVRR.COPYBOOK

Member	Description
@INDEX.C	CONNX VSAM Sample File/COBOL Copybook XREF
CUSTOMER.C	Sample Customer KSDS
CUSTOMRE.C	Sample Customer ESDS
CUSTOMRR.C	Sample Customer RRDS
EQUIPMNE.C	Sample Equipment ESDS
EQUIPMNR.C	Sample Equipment RRDS
EQUIPMNT.C	Sample Equipment KSDS
ORDER.C	Sample Order KSDS
ORDERE.C	Sample Order ESDS
ORDERR.C	Sample Order RRDS
PRODGRP.C	Sample Product Group KSDS
PRODGRPE.C	Sample Product Group ESDS
PRODGRPR.C	Sample Product Group RRDS
PRODUCT.C	Sample Product KSDS
PRODUCTE.C	Sample Product ESDS
PRODUCTR.C	Sample Product RRDS

CNXVVRR.CNTL

Member	Description
REMOVE0.JCL	CONNX for VSAM/VSE Complete Uninstall
REMOV10.JCL	CONNX for VSAM/VSE Partial Uninstall
CNXINST0.JCL	CONNX for VSAM/VSE Installation
CNXINS10.JCL	CONNX for VSAM/VSE JCL for CICS V2R3 and below partitions

CNXVVRR.OBJLIB

Member	Description
CNXCFCG.OBJ	CONNX CICS/VSE configuration program object code
CNXRUN.OBJ	CONNX CICS/VSE TCP/IP listener program object code
CNXSTOP.OBJ	CONNX CICS/VSE Automated TCP/IP Listener shutdown program object code

CNXVSD0.OBJ	CONNX CICS/VSE TCP/IP server program object code
CNXSIGN.OBJ	CONNX CICS/VSE 2.3 External Security Interface program object code
VSOCKET.OBJ	CONNX CICS/VSE TCP/IP object code stub for socket() function

CNXVRR.RUNLIB

Member	Description
CNXCFC.PHASE	CONNX CICS/VSE configuration program PHASE
CNXRUN.PHASE	CONNX CICS/VSE TCP/IP listener program PHASE
CNXSTOP.PHASE	CONNX CICS/VSE Automated TCP/IP listener shutdown program PHASE
CNXVSD0.PHASE	CONNX CICS/VSE TCP/IP server program PHASE

CNXVRR.SOURCE

Member	Description
CNXFCT0.ASM	File Control Table source for CONNX Configuration and Sample VSAM Files for CICS/VSE 2.4 and above
CNXPPT0.ASM	Program Processing Table source for CONNX programs
SASDCT.ASM	SAS/C Destination Control Table source
SASPPT.ASM	SAS/C Program Processing Table source
V2R3FCT0.ASM	CICS V2R3 and below file control table source for CONNX sample VSAM files

Part 4: CONNX Configuration Utility

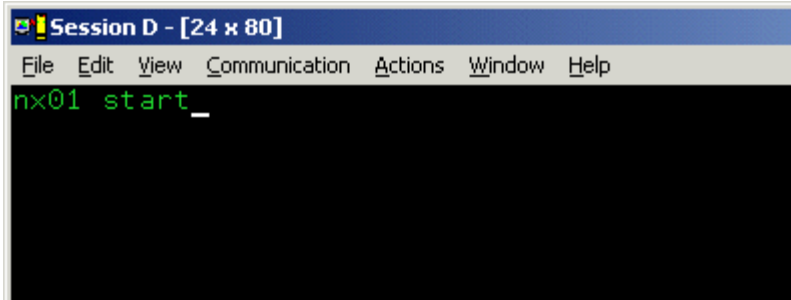
Configuration Utility

The CONNX configuration utility for VSAM/CICS enables the user to start the TCP/IP listener, and to set, remove, and display current CONNX variables.

To start the CONNX CICS/VSAM TCP/IP Listener

The listener process waits for and verifies connections with client machines.

1. After logging on to the CICS region, enter **NX01 Start**, and then press **<Enter>**.

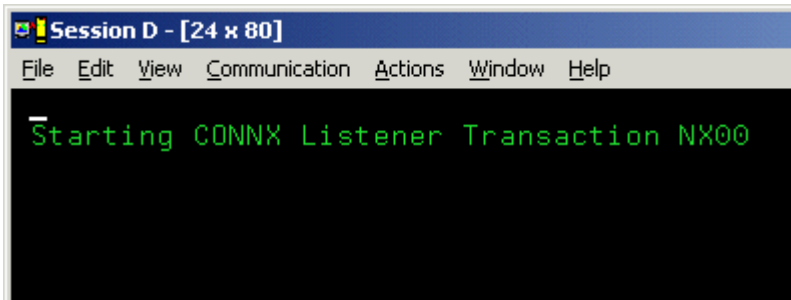


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
nx01 start_

```

2. The CONNX TCP/IP Listener starts.

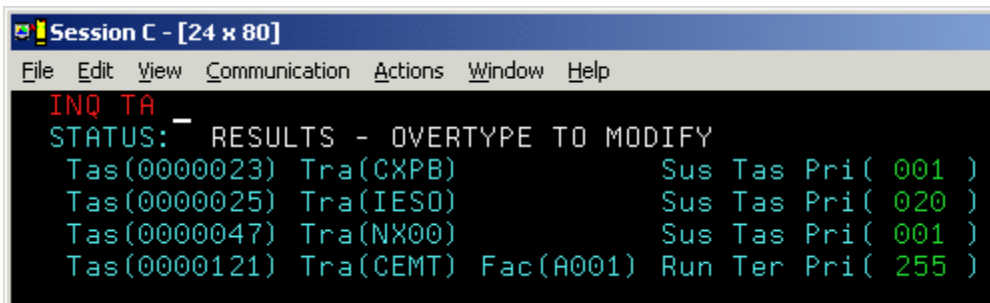


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
Starting CONNX Listener Transaction NX00

```

3. Verify start by typing **CEMT INQ TA**. Press **<Enter>**. The screen displays a list of transactions currently running. Verify that **Tas(0000xxx) Tra(NX00)** is listed as one of the transactions.



```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
INQ TA
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CXPB) Sus Tas Pri( 001 )
Tas(0000025) Tra(IES0) Sus Tas Pri( 020 )
Tas(0000047) Tra(NX00) Sus Tas Pri( 001 )
Tas(0000121) Tra(CEMT) Fac(A001) Run Ter Pri( 255 )

```

4. For **CICS/VSE 2.3** and below installations, starting and stopping the CONNX CICS/VSAM TCP/IP Listener requires a dedicated terminal for the CNXSIGN external security interface program. For example, if NX01 START is issued from a CICS/VSE 2.3 terminal session, followed by a CEMT INQ TA, the terminal will lock until the CONNX Listener transaction is stopped with the NX01 STOP command. In the following example, the NX01 STOP command is issued from the VSE Console, followed by CEMT INQ TA:

Example 1:

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help

DFH3504 15:40:08 SIGNON IS COMPLETE (LANGUAGE E).
nx01 start_

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help

Starting CONNX Listener Transaction NX00

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help

cemt inq ta_

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

SYSTEM: VSE/ESA VSE/ESA 2.4.0 TURBO (01) USER: CTCM
TIME: 15:42:42 CTCM
F4-0004
----- E N D O F F I L E -----

==> 4 nx01 stop_

1=HLP 2=CPY 3=END 6=CNCL 7=BWD 8=FWD 9=EXPL 10=INP 12=INFO
FILTER: ALL FWD MODE: REDISPLAY
MA a 20/017

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
SYSTEM: VSE/ESA VSE/ESA 2.4.0 TURBO (01) USER: CTCM
TIME: 15:44:11
Y1 0001 1Q47I Y1 DITECTCM 01508 FROM (CTCM) , TIME=15:06:15
Y1 0045 // JOB DITECTCM DITTO/FS
DATE 10/24/2002, CLOCK 15/06/15
F4-0004
4 nx01 stop
F4 0004
F4 0004 Stopping CONNX Listener Transaction NX00 on port 6600.
F4 0004 CONNX Listener Transaction NX00 Stopped
F4 0004
F4 0004
F4 0004
F4-0004

==> _
1=HLP 2=CPY 3=END 4=RTN 5=DEL 6=DELS 7=RED 8=CONT 9=EXPL 10=HLD 12=RTRV
ACT_MSG: HOLDRUN PAUSE: 01 SCROLL: 1 MODE: CONSOLE
MA a 20/006

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
inq ta
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(00015) Tra(CXPB) Act Tas
Tas(00062) Tra(CEMT) Fac(A001) Act Ter

```

Example 2:

Alternatively, the CONNX CICS TCP/IP Listener can be started from the VSE console and stopped from a CICS/VSE 2.3 session:

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
SYSTEM: VSE/ESA VSE/ESA 2.4.0 TURBO (01) USER: CTCM
TIME: 15:44:11
Y1 0001 1Q47I Y1 DITECTCM 01508 FROM (CTCM) , TIME=15:06:15
Y1 0045 // JOB DITECTCM DITTO/FS
DATE 10/24/2002, CLOCK 15/06/15
F4-0004
4 nx01 stop
F4 0004
F4 0004 Stopping CONNX Listener Transaction NX00 on port 6600.
F4 0004 CONNX Listener Transaction NX00 Stopped
F4 0004
F4 0004
F4 0004
F4-0004

==> 4 nx01 start_

1=HLP 2=CPY 3=END 4=RTN 5=DEL 6=DELS 7=RED 8=CONT 9=EXPL 10=HLD 12=RTRV

ACT_MSG: HOLDRUN PAUSE: 01 SCROLL: 1 MODE: CONSOLE
Mâ a 20/018

```

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
SYSTEM: VSE/ESA VSE/ESA 2.4.0 TURBO (01) USER: CTCM
TIME: 15:52:59
DATE 10/24/2002, CLOCK 15/06/15
F4-0004
4 nx01 stop
F4 0004
F4 0004 Stopping CONNX Listener Transaction NX00 on port 6600.
F4 0004 CONNX Listener Transaction NX00 Stopped
F4 0004
F4 0004
F4 0004
F4 0004 Starting CONNX Listener Transaction NX00
F4 0004
F4 0004
F4 0004
F4 0004
==> _
1=HLP 2=CPY 3=END 6=CNCL 7=BWD 8=FWD 9=EXPL 10=INP 12=INFO
FILTER: ALL BWD MODE: REDISPLAY
MA a 20/000

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
inq ta
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(00015) Tra(CXPB) Act Tas
Tas(00064) Tra(NX00) Fac(CNSL) Act Ter
Tas(00062) Tra(CEMT) Fac(A001) Act Ter

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
nx01 stop
STATUS: SESSION ENDED

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help

Stopping CONNX Listener Transaction NX00 on port 6600.
CONNX Listener Transaction NX00 Stopped

```

```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
inq ta
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(00015) Tra(CXPB) Act Tas
Tas(00069) Tra(CEMT) Fac(A001) Act Ter

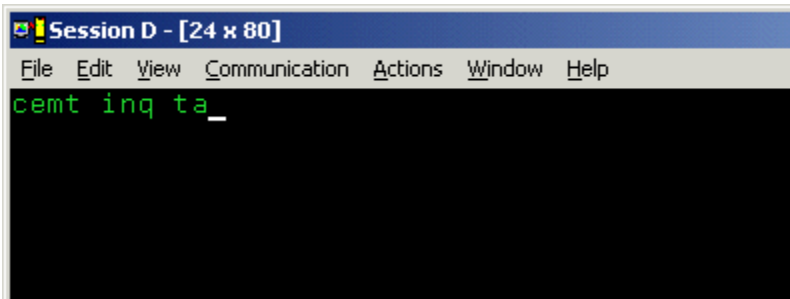
```

In Example 2, the NX01 START command is issued from the VSE console, which creates an NX00 transaction with a principal facility terminal of CNSL. The NX00 transaction can also be started from an autoinstalled batch terminal. In all cases, the invoking terminal is dedicated to transaction NX00 until it completes. In the current example, additional messages from the VSE console to partition F4 are deferred until transaction NX00 completes.

To stop the CONNX CICS/VSAM TCP/IP Listener

You can stop the CONNX CICS/VSAM TCP/IP listener program with the NX01 transaction from an authorized 3270 terminal session. The NX01 STOP command gracefully terminates the CONNX CICS/VSAM TCP/IP listener, and permits it to be restarted on the same or on a different TCP/IP port.

1. Verify that the CONNX CICS TCP/IP listener transaction/program (NX00/CNXRUN) is running by typing **cemt inq ta**. Press **<Enter>**.

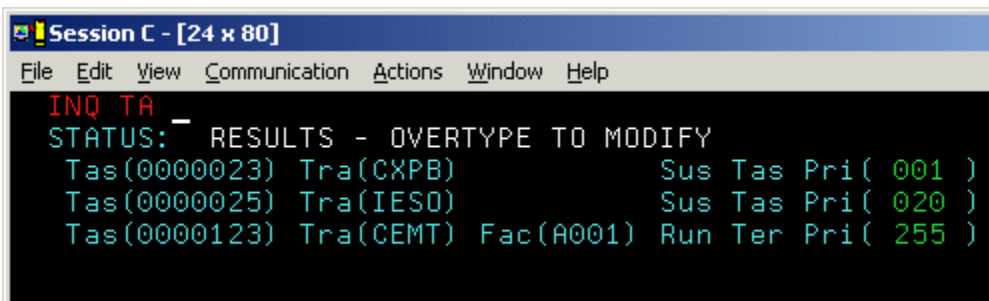


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
cemt inq ta_

```

2. Verification appears. Press the **<F3>** key, and then press **<Clear>**.

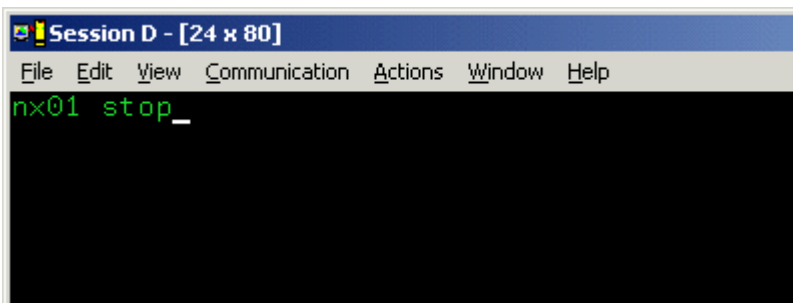


```

Session C - [24 x 80]
File Edit View Communication Actions Window Help
INQ TA
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CXPB)          Sus Tas Pri( 001 )
Tas(0000025) Tra(IES0)          Sus Tas Pri( 020 )
Tas(0000123) Tra(CEMT) Fac(A001) Run Ter Pri( 255 )

```

3. Stop the CONNX CICS TCP/IP listener by typing **nx01 stop**, and then press **<Enter>**.

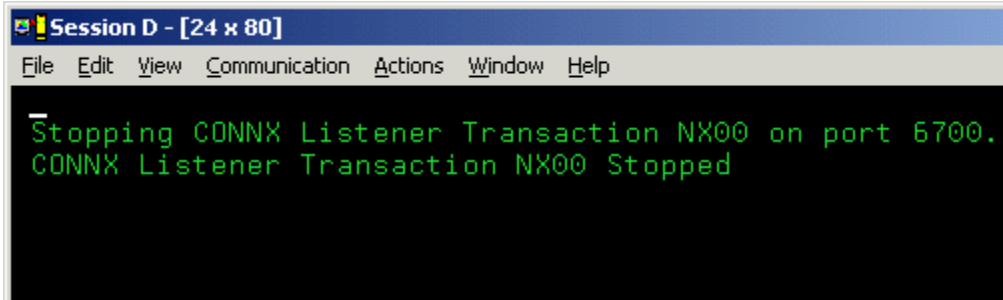


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
nx01 stop_

```

4. Verification appears.



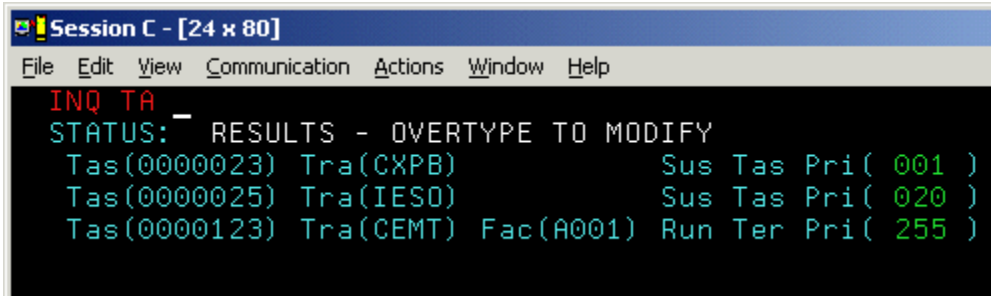
```

Session D - [24 x 80]
File Edit View Communication Actions Window Help

Stopping CONNX Listener Transaction NX00 on port 6700.
CONNX Listener Transaction NX00 Stopped

```

5. You can verify further that the CONNX CICS TCP/IP listener transaction/program (NX00/CNXRUN) is stopped by issuing a second **CEMT INQ TA** command, which shows that the NX00 task has been stopped. You can restart the CONNX CICS/VSAM TCP/IP Listener program on the same TCP/IP port, or you can change the TCP/IP port using the NX01 CNXRUNPORT command, and restart the listener on a different port.



```

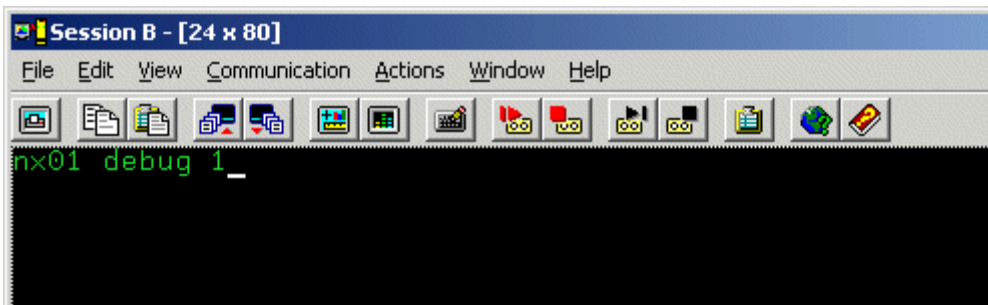
Session C - [24 x 80]
File Edit View Communication Actions Window Help

INQ TA
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CXPB)          Sus Tas Pri( 001 )
Tas(0000025) Tra(IES0)          Sus Tas Pri( 020 )
Tas(0000123) Tra(CEMT) Fac(A001) Run Ter Pri( 255 )

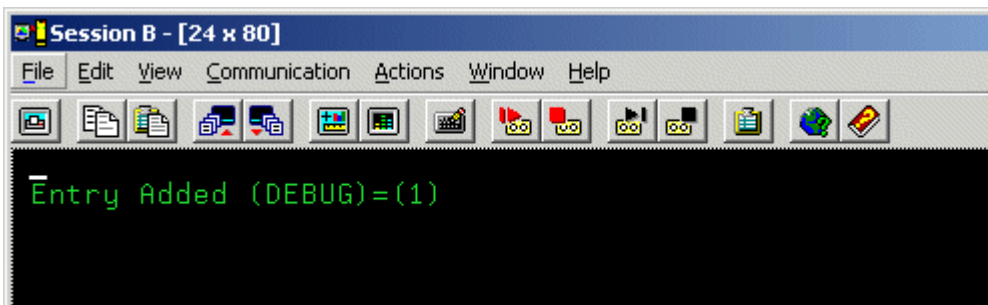
```

To set a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE> <VALUE>**, and then press **<Enter>** to turn on the CONNX variable function. In this example, the DEBUG variable is used.

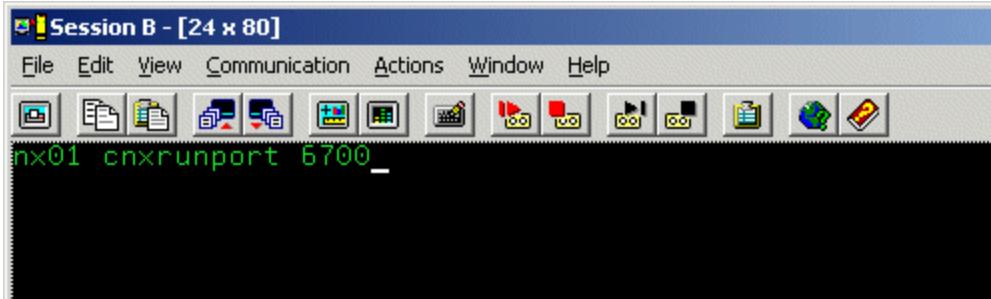


2. The variable is set, and verification appears.

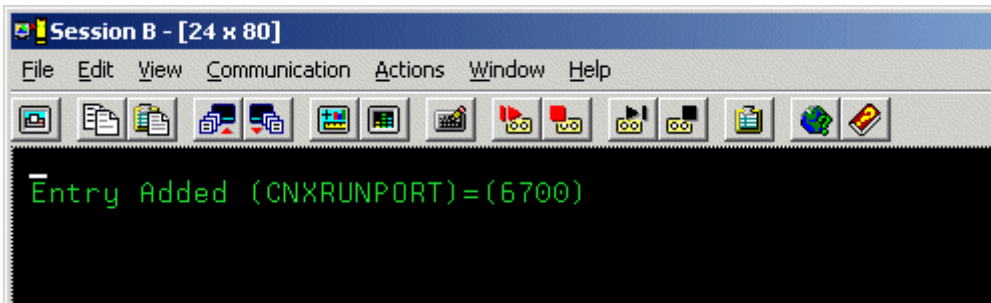


To change the number of the CONNX TCP/IP port

1. Clear the screen, type **NX01 CNXRUNPORT <NewPortNumber>**, and then press **<Enter>**. (The default CONNX port is 6500.) The new port number used in this example is 6700.
2. If you change this port, all clients must be updated to the same port number. See TCP/IP Connection Port in the *CONNX User Reference Guide*, available online, on the CONNX CD-ROM, and within the CONNX product, for more information on changing the TCP/IP connection port for client machines.

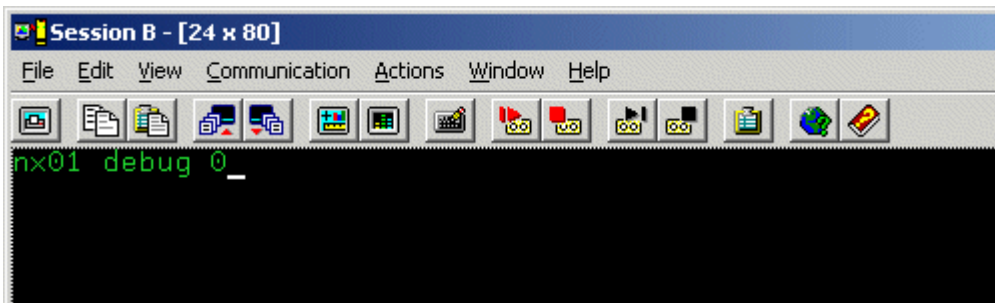


3. The port number is changed, and verification appears.

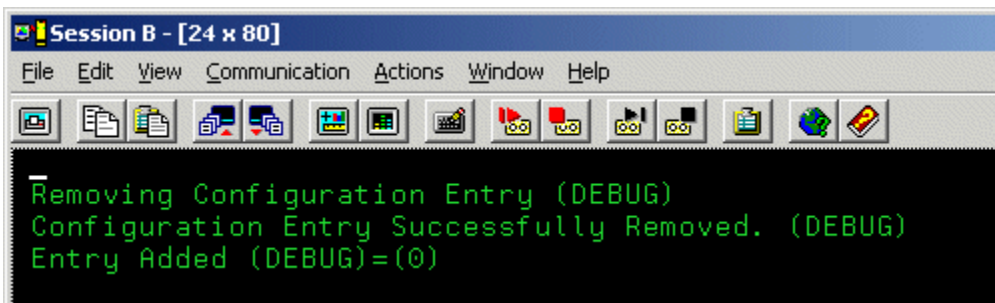


To change a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE>**, and then press **<Enter>**. In this example, in which the variable was set in [To set a CONNX environment variable](#), the **DEBUG** variable is used.

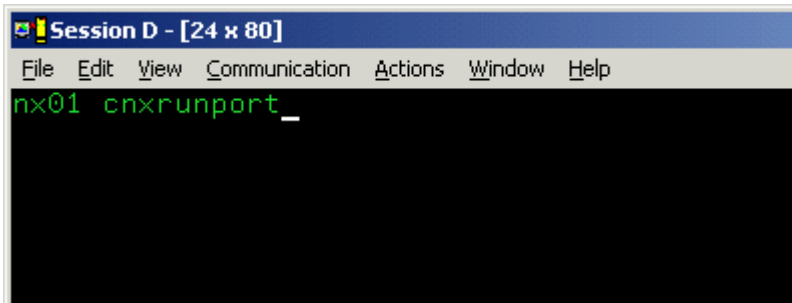


2. The variable is removed, and verification appears.

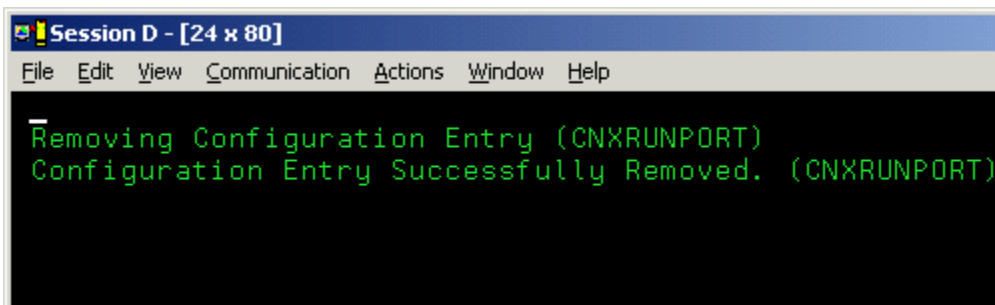


To convert the CONNX port number to the default

1. Clear the screen, type **NX01 CNXRUNPORT**, and then press **<Enter>**.



2. The current entry for CNXRUNPORT is removed, and verification appears. Removal of the CNXRUNPORT variable restores the default value of 6500.



To control VSAM file Enables/Opens with CONNX Environment Variables SETENABLE and SETOPEN

CONNX for VSAM provides two environment variables which can be used to control file ENABLE and OPEN requests. For example, CICS VSAM files are usually defined to RDO (Resource Definition Online) with initial statuses:

STATUS(UNENABLED) OPENTIME(FIRSTREF)

or to the FCT (File Control Table) via the macro syntax:

FILSTAT=(ENABLED,CLOSED)

When the CONNX VSAM server program receives an SQL request from an ODBC/OLE DB/.NET/JDBC application, it first checks the status of the target VSAM file. If the VSAM file is CLOSED and not currently ENABLED, the CONNX VSAM server program sets the VSAM file status to ENABLED and OPEN, and then performs the SQL request. This default behavior can be overridden by two CONNX environment variables: SETENABLED and SETOPEN.

The CONNX SETENABLED and SETOPEN environment variables can be added, changed, and deleted via the NX01 CICS Transaction. The valid values for both SETENABLED and SETOPEN are 0 = off and 1 = on. The default value for both SETENABLED and SETOPEN is 1.

The possible combinations and actions for SETENABLED and SETOPEN are:

SETENABLED	SETOPEN	Action
0	0	If the file is not ENABLED, or if the file is ENABLED but not OPEN, reject the query.
0	1	If the file is not ENABLED, reject the query. If the file is ENABLED but not OPEN, open the file and execute the query.
1	0	If the file is not ENABLED, ENABLE the file. If the file is not OPEN, reject the query.
1	1	Default Settings. If the file is not ENABLED, ENABLE the file. If the file is not OPEN, open the file and execute the query.

In all of the cases above, if the target VSAM file is ENABLED and OPEN, execute the query.

1. To set the CONNX SETENABLED and SETOPEN environment variables, logon to the CICS region / partition on which CONNX is installed, clear the screen, and enter:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
nx01 SETENABLED 1_
    
```

- The **Entry Added** message appears.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Entry Added (SETOPEN)=(1)
    
```

- To see the CONNX SETOPEN environment variable, logon to the CICS region/partition on which CONNX is installed, clear the screen, and enter **nx01 SETOPEN 1**.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
nx01 SETOPEN 1_
    
```

- The **Entry Added** message appears.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Entry Added (SETOPEN)=(1)
    
```

- To display the current CONNX environment variables, clear the screen and enter NX01:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Displaying CONNX Configuration for CICS APPLID = CICS
(CNXRUNPORT)=(6700)
(SETENABLED)=(1)
(SETOPEN)=(1)
    
```

Examples:

6. The default case. The CONNX sample VSAM KSDS file CNXCSTK has not been accessed by a CONNX query. The initial file status as displayed by the CEMT INQ FI(CNXCSTK) transaction is CLOSED (Clo) and UNENABLED (Une).

The screenshot shows a terminal window titled "Session B - [24 x 80]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The main display area shows the command "INQ FI(CNXCSTK)" in red. Below it, the status is displayed as "STATUS: RESULTS - OVERTYPE TO MODIFY". The file status is shown as "Fil(CNXCSTK) Vsa Clo Une Rea Upd Add Bro Del", where "Clo" and "Une" are highlighted in green. At the bottom, the response is "RESPONSE: NORMAL". The system ID and application ID are "SYSID=CIC2 APPLID=PRODCICS". The time and date are "TIME: 13.06.28 DATE: 04.18.03". The bottom status bar shows "PF 1 HELP 3 END" and "7 SBH 8 SFH 9 MSG 10 SB 11 SF". The bottom right corner shows "01/019".

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
INQ FI(CNXCSTK)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(CNXCSTK ) Vsa Clo Une Rea Upd Add Bro Del

RESPONSE: NORMAL
SYSID=CIC2 APPLID=PRODCICS
TIME: 13.06.28 DATE: 04.18.03
PF 1 HELP 3 END 7 SBH 8 SFH 9 MSG 10 SB 11 SF
01/019

```

7. If the CONNX SETENABLED and SETOPEN environment variables have not been defined with the NX01 transaction, or if they are defined and set to the default value = 1 as above, then the first CONNX query against the CNXCSTK file will cause the file to be ENABLED and OPENED before the query results are returned to the requesting client. The file remains ENABLED and OPENED for the lifetime of the CICS region/partition. The file status after successful query execution is:


```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
INQ FI(CNXCSTK)
STATUS: RESULTS- OVERTYPE TO MODIFY
Fil(CNXCSTK ) Vsa Ope Ena Rea Upd Add Bro Del

RESPONSE: NORMAL
TIME: 13.08.24 DATE: 04.18.03
PF 1 HELP 3 END 7 SBH 8 SFH 9 MSG 10 SB 11 SF
MA b 01/019

```

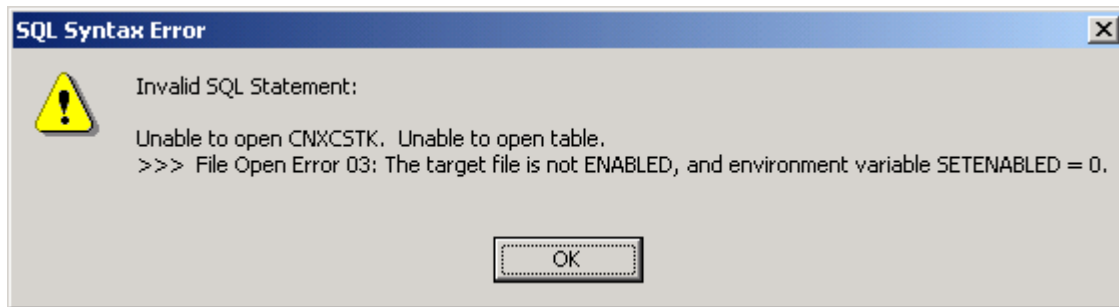
8. Both SETENABLED and SETOPEN are defined and set to 0.

```

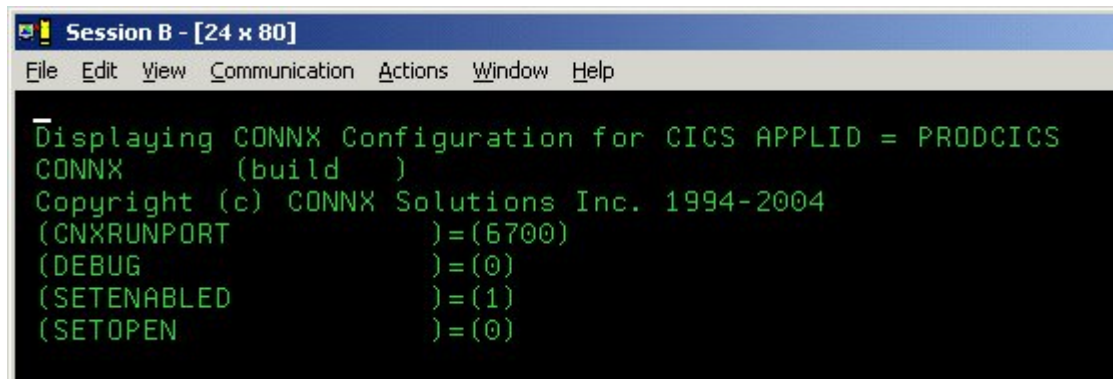
Session B - [24 x 80]
File Edit View Communication Actions Window Help
Displaying CONNX Configuration for CICS APPLID = PRODCICS
CONNX (build: )
Copyright (c) CONNX Solutions Inc. 1994-2004
(CNXRUNPORT)=(6700)
(DEBUG)=(0)
(SETENABLED)=(0)
(SETOPEN)=(0)

```

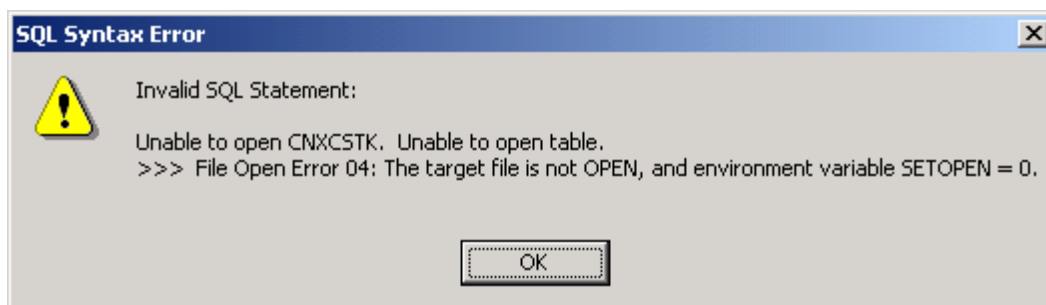
9. The CNXCSTK VSAM file is in its initial state of CLOSED (Clo) and UNENABLED (Une). When the CONNX VSAM server program receives a query request for the CNXCSTK file, it checks the current status and finds that the file is not ENABLED and CLOSED. The CONNX variable SETENABLED is defined as 0, so the CONNX VSAM server bypasses the file enable logic and returns the following message to the client application:



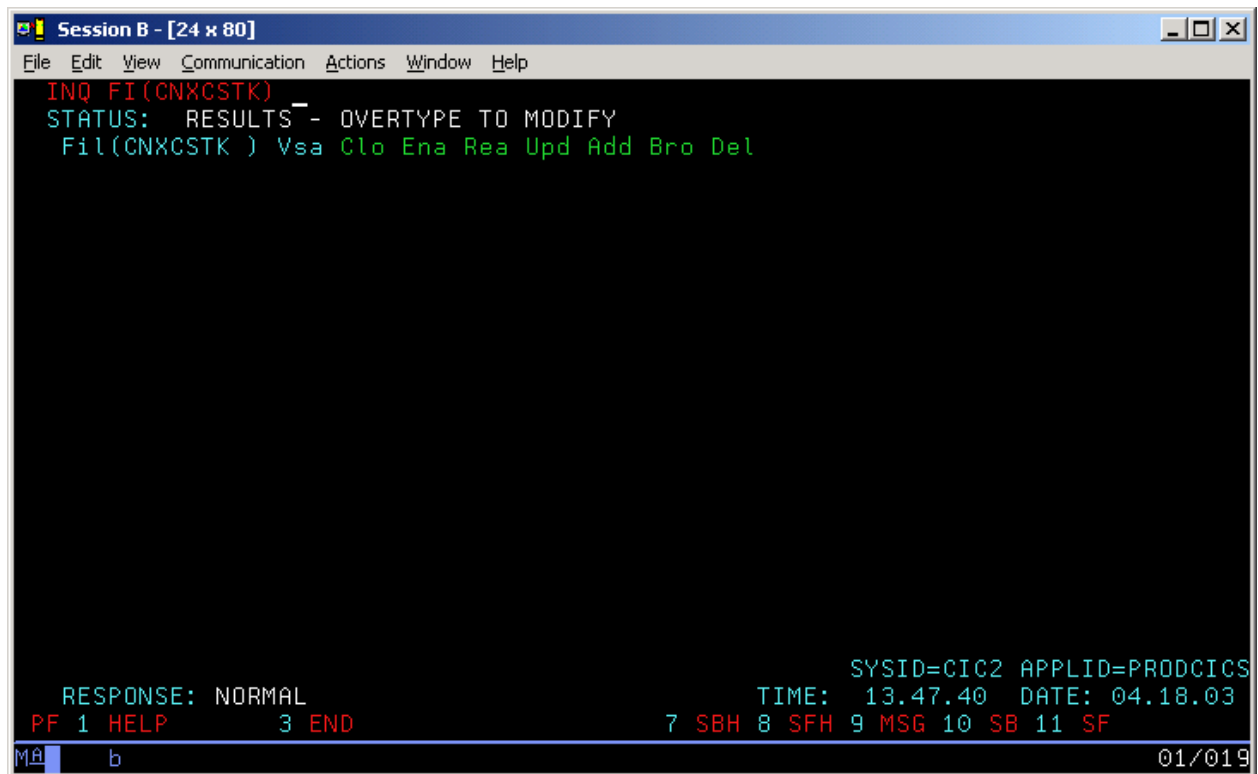
10. SETENABLED is set to 1 and SETOPEN to 0.



11. The CNXCSTK VSAM file is in its initial state of CLOSED (Clo) and UNENABLED (Une). When the CONNX VSAM server program receives a query request for the CNXCSTK file, it checks the current status and finds that the file is not ENABLED and CLOSED. The CONNX VSAM server program enables the file and bypasses the open attempt, because SETOPEN is set to 0. The following message is returned to the client application:



12. The CNXCSTK VSAM file status is now ENABLED and CLOSED:



```
Session B - [24 x 80]
File Edit View Communication Actions Window Help
INQ FI(CNXCSTK)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(CNXCSTK ) Vsa Clo Ena Rea Upd Add Bro Del

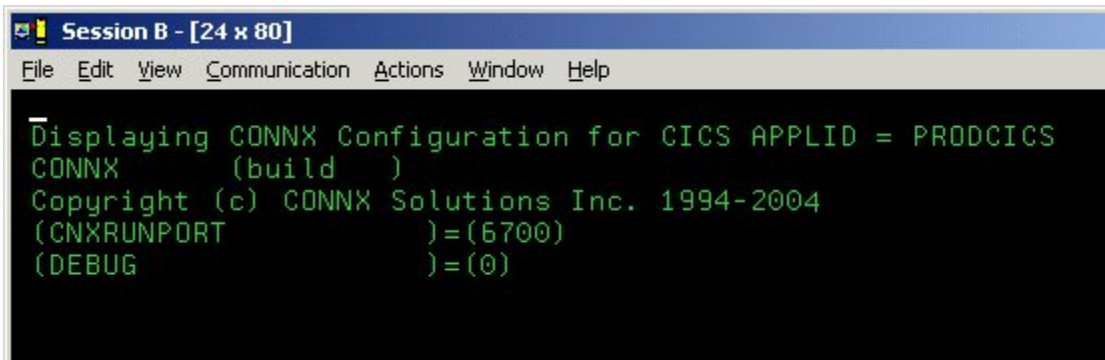
RESPONSE: NORMAL
PF 1 HELP 3 END
SYSID=CIC2 APPLID=PRODCICS
TIME: 13.47.40 DATE: 04.18.03
7 SBH 8 SFH 9 MSG 10 SB 11 SF
MA b 01/019
```

To enable the Barnard TCP/IP Stack

1. Clear the screen, type **NX01 cnxbarnard 1**, and then press **<Enter>** to turn on the CONNX variable function that enables the Barnard TCP/IP stack.
2. The variable is set, and verification appears.

To display the current environment variable(s)

1. Type **NX01**, and then press **<Enter>** to display the current configuration.



```
Session B - [24 x 80]
File Edit View Communication Actions Window Help

Displaying CONNX Configuration for CICS APPLID = PRODCICS
CONNX      (build      )
Copyright (c) CONNX Solutions Inc. 1994-2004
(CNXRUNPORT      )=(6700)
(DEBUG          )=(0)
```

2. To test your connection, start the CONNX Data Dictionary Manager application and follow the import procedures described in CONNX and VSAM in the CONNX User Reference Guide available online, on the CONNX CD-ROM, and within the CONNX product.

Part 5: CONNX for VSAM - VSE: Uninstallation Procedures

Uninstallation Procedures

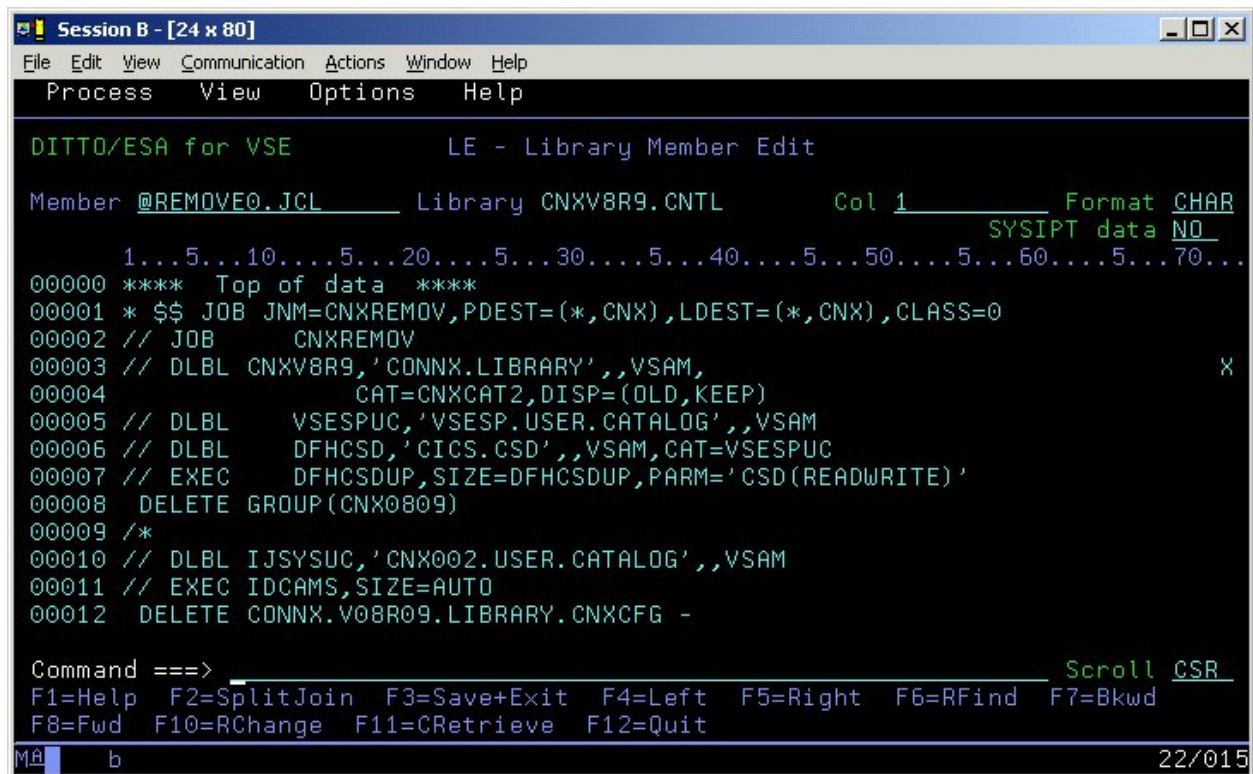
The CONNX for VSAM TCP/IP Server Setup client application transfers the CONNX components to the target VSE host via FTP. After the initial FTP step, the CONNX installation continues via a batch job submission using VSE Ditto. The final installation steps on VSE platforms consist of manual updates to the CICS target partition Destination Control Table and the startup JCL.

CONNX for VSAM provides batch uninstallation JCL for VSE platforms.

To uninstall CONNX for VSAM - VSE

Given a default CONNX installation library name of CNXVVRR, the CONNX installation JCL library.sublib is CNXVVRR.CNTL. Member @REMOVE0.JCL contains the batch uninstallation JCL.

1. Log on to the target CICS partition and stop the CONNX NX00 TCP/IP listener transaction NX00 by entering:
NX01 STOP
2. Next, shut down CICS by entering:
CEMT PERF SHUT IMMED
3. After the CICS partition shuts down, you can submit the JCL in member @REMOVE0.JCL via the CICS Ditto transaction. Note that the first control record of the // EXEC LIBR step deletes the CNXVVRR.CNTL library.sublib, so be sure to submit the job and deallocate the CONNX library by exiting from the CICS Ditto transaction.

Example:


```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 * $$ JOB JNM=CNXREMOV,PDEST=(*,CNX),LDEST=(*,CNX),CLASS=0
00002 // JOB CNXREMOV
00003 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM, X
00004 CAT=CNXCAT2,DISP=(OLD,KEEP)
00005 // DLBL VSESPUC,'VSESP.USER.CATALOG',,VSAM
00006 // DLBL DFHCSD,'CICS.CSD',,VSAM,CAT=VSESPUC
00007 // EXEC DFHCSDUP,SIZE=DFHCSDUP,PARM='CSD(READWRITE)'
00008 DELETE GROUP(CNX0809)
00009 /*
00010 // DLBL IJSYSUC,'CNX002.USER.CATALOG',,VSAM
00011 // EXEC IDCAMS,SIZE=AUTO
00012 DELETE CONNX.V08R09.LIBRARY.CNXCFC -

Command ==> Scroll CSR
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```

4. The IDCAMS DELETE step should return zeroes for all CONNX data sets which exist and are not allocated to another VSE process or job. The EXEC LIBR step to delete the CONNX VSAM-managed library and sublibraries may return an informational message to the effect that the deletes should be performed with Access Method Services. The CICS RDO DFHCSDUP utility step should return zeroes for deletion of the CNX0809 group. The final IDCAMS DELETE step removes the VSAM cluster which contains the CNXV8R8 library. The last IESVCLUP step removes the standard label for the CNXVVRR library, if one was created during installation.

Manual Steps: CONNX for VSE Uninstallation

The startup JCL for the CICS partition on which CONNX has been installed has been manually updated to concatenate the CONNX application library.sublib to the LIBDEF search list. The references to CNXVRRR must be deleted.

Example:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CICS2.Z             Library IJSYSRS.SYSLIB      Col 1          Format CHAR
                               SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 $$$$ JOB JNM=CICS2,DISP=L,CLASS=8,EJMSG=YES
00002 $$$$ LST CLASS=A,DISP=D,RBS=100
00003 // JOB CICS2          STARTUP OF SECOND CICS WITHOUT ICCF
00004 // OPTION SADUMP=5
00005 // OPTION SYSDUMPC
00006 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM,CAT=CNXCAT2
00007 // DLBL CNXCICS,'CONNX.CICS.LIBRARY',,VSAM,CAT=CNXCAT2
00008 // LIBDEF *,SEARCH=(CNXCICS.CICS,                      X
00009                      CNXV8R9.RUNLIB,                      X
00010                      PRD2.SERV140E,                        X
00011                      PRD2.TCPIP,                          X
00012                      PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD, X

Command ==>                               Scroll CSR
F1=Help  F2=SplitJoin F3=Save+Exit F4=Left  F5=Right F6=RFind F7=Bkwd
F8=Fwd   F10=RChange F11=CRetrieve F12=Quit

MA  b                                     22/015

```

Additionally, three SAS/C-specific DLBLs (STDERR, STDOUT, and STGRPT) reference sequential datasets dedicated to the CICS extra-partition Temporary Data Queues SASE, SASO, and SASR. These data set references should be removed from the CICS Startup JCL:

Example:


```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CICS2.Z             Library IJSYSRS.SYSLIB      Col 1          Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00026 NOLOG
00027 // EXEC PROC=DTRCICS2          LABELS FOR CICS FILES
00028 // ASSGN SYS012,DISK,VOL=CNX002,SHR
00029 // DLBL STDOUT,'CONNX.SASC700.PRODCICS.STDOUT',0,SD
00030 // EXTENT SYS012,CNX002,1,0,12993,75
00031 // DLBL STDERR,'CONNX.SASC700.PRODCICS.STDERR',0,SD
00032 // EXTENT SYS012,CNX002,1,0,13068,5
00033 // DLBL STGRPT,'CONNX.SASC700.PRODCICS.STGRPT',0,SD
00034 // EXTENT SYS012,CNX002,1,0,13073,15
00035 *   WAITING FOR VTAM TO COME UP
00036 // EXEC IESWAITT
00037 $$/*
00038 // IF XMODEF8 = COLD THEN

Command ==>
F1=Help  F2=SplitJoin  F3=Save+Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd
F8=Fwd   F10=RChange  F11=CRetrieve  F12=Quit

MA b                                     22/015

```

Finally, you can delete the sequential SAS TDQ datasets via the CICS Ditto Utility.

Running CONNX for VSE Uninstallation Without Shutting Down CICS

You can perform a partial uninstallation of CONNX without shutting down the target CICS partition as follows:

1. Log on to the target CICS partition and stop the CONNX CICS TCP/IP Listener transaction NX00 by entering:
NX01 STOP
2. Next, use the CEMT transaction to close and disable the CONNX VSAM configuration and sample files.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
INQ FI(CNX*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(CNXCFG ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXCSTE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXCSTK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXCSTR ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXEQI1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQI2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQP1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQP2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQR ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXORI1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORI2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORP1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORP2 ) Vsa Ope Ena Rea Upd Add Bro Del
+ Fil(CNXORR ) Vsa Ope Ena Rea Upd Add Bro Del

RESPONSE: NORMAL
TIME: 14.08.31 DATE: 04.18.03
PF 1 HELP 3 END 7 SBH 8 SFH 9 MSG 10 SB 11 SF
MA b 01/016

```

3. Edit the @REMOVE0 JCL member and delete the final three steps beginning with:

```
// EXEC LIBR,PARM='MSHP'
```

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00091 // EXEC LIBR,PARM='MSHP'
00092      DELETE SUBLIB=CNXV8R9.CICS
00093      DELETE SUBLIB=CNXV8R9.CNTL
00094      DELETE SUBLIB=CNXV8R9.COPYBOOK
00095      DELETE SUBLIB=CNXV8R9.OBJLIB
00096      DELETE SUBLIB=CNXV8R9.RUNLIB
00097      DELETE SUBLIB=CNXV8R9.SOURCE
00098 /*
00099 // EXEC IDCAMS,SIZE=AUTO
00100      DELETE CONNX.LIBRARY
00101      CLUSTER PURGE
00102      CATALOG (CNX002.USER.CATALOG)
00103 /*

Command ==>
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```

4. Save and submit the modified JCL via the CICS Ditto transaction.
5. After this job completes and the target CICS partition is shut down, start another CICS Ditto session and submit JCL member @REMOV10 to delete the CNXV8R8 library and the standard label. Follow the manual steps above to complete the CONNX uninstallation.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00091 // EXEC LIBR,PARM='MSHP'
00092      DELETE SUBLIB=CNXV8R9.CICS
00093      DELETE SUBLIB=CNXV8R9.CNTL
00094      DELETE SUBLIB=CNXV8R9.COPYBOOK
00095      DELETE SUBLIB=CNXV8R9.OBJLIB
00096      DELETE SUBLIB=CNXV8R9.RUNLIB
00097      DELETE SUBLIB=CNXV8R9.SOURCE
00098 /*
00099 // EXEC IDCAMS,SIZE=AUTO
00100      DELETE CONNX.LIBRARY
00101      CLUSTER PURGE
00102      CATALOG (CNX002.USER.CATALOG)
00103 /*

Command ==>
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```


Adabas SQL Gateway (CONNX for Adabas) - VSE

Prerequisites for Adabas SQL Gateway - VSE

The IBM VSE target system must be configured to support TCP/IP communications software from CSI international, IBM, or Barnard Software, Inc.

For VSE installations, the only other setup step required is to concatenate the VSE TCP/IP library to the target CICS partition LIBDEF search list. For example, given the default VSE TCP/IP, CONNX and IBM Language Environment library names:

```
// LIBDEF *,SEARCH=(CNXVVR.RUNLIB,          X
PRD2.TCPIP,                                X
PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD, X
PRD2.SCEECICD,PRD2.SCEECICS,PRD2.SCEEBASD,  X
PRD2.SCEEBASE,PRD2.DBASE),PERM
```

Verifying TCP/IP Connectivity

Before you install CONNX, we recommend that you verify the TCP/IP connectivity to your VSE system. To verify connectivity:

1. Open a Command Prompt window.
2. On the task bar, click **Start** and then click **Run**
3. Type or select:

ping hostname

where **hostname** is the TCP/IP symbolic address or dotted numeric address.

A successful ping (connectivity) will generate output in the command window similar to the following:

```
Pinging 123.123.123.012 with 32 bytes of data:
Reply from 123.123.123.012: bytes=32 time=10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Reply from 123.123.123.012: bytes=32 time<10ms TTL=63
Ping statistics for 123.123.123.012:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Part 1: Creating the CONNX Library in Adabas-managed space

Creating the CONNX Library

The Adabas SQL Gateway provides an automated installation client application that transfers the CONNX server components to the target VSE host via TCP/IP FTP. To succeed, the automated FTP step requires a target VSE library defined in VSAM-managed space. The target library must be defined manually, either via the following CONNX-supplied template and FTP, or by using the VSE resource-definition panels. To use the CONNX installation template, follow the steps below:

1. The CONNX VSE installation template is copied to the CONNX VSE installation directory on the client machine; the default name is

C:\connx32\adabas\templates\ftp\vse\cnxlibv_.dat

2. Using your editor of choice, such as MS Notepad, Wordpad, Write, etc., open the cnxlibv_.dat file and make the following substitutions for the template variables defined in the following table:

Adabas SQL Gateway (CONNX for Adabas) - VSE Installation Template Substitutable Variables

Variable	Definition
[CNXLIB]	The 1-7 character Adabas SQL Gateway library name defined in the [CNXLIBP] CONNX library cluster; the default is CNXVVR.
[CNXLIBP]	The 1-44 character fully-qualified VSE cluster name which contains the CONNX Adabas SQL Gateway library; the default is CONNX..LIBRARY.
[CNXCATL]	The 1-7 character VSE standard label which references the physical data set name of the Adabas SQL Gateway user or system catalog ([CNXCATP]).
[CNXCATP]	The 1-to-44 character physical data set name of the VSE Adabas SQL Gateway user catalog into which the Adabas SQL Gateway-managed library [CNXLIBP] is defined. If left blank, the CONNX library is defined in the default VSE system catalog.
[SAMPLEVOLSER]	The 1-6 character DASD volume serial number on which the VSE Adabas SQL Gateway-managed library is installed: no default.

A sample edit session appears below:

```

cnxlibv$ - Notepad
File Edit Format Help
* $$$ JOB JNM=CNXLIB,CLASS=0,DISP=D,NTFY=YES
// JOB CNXLIB DEFINE CONNX for VSAM/VSE VSAM-managed library
// EXEC LIBR,PARM='MSHP'
//      DELETE SUBLIB=[CNXLIB].CNTL
//      DELETE SUBLIB=[CNXLIB].COPYBOOK
//      DELETE SUBLIB=[CNXLIB].OBJLIB
//      DELETE SUBLIB=[CNXLIB].RUNLIB
//      DELETE SUBLIB=[CNXLIB].SOURCE
//
// EXEC IDCAMS,SIZE=AUTO
//      DELETE [CNXLBP]          -
//      CLUSTER PURGE          -
//      CATALOG ([CNXCATP])
//
// OPTION STDLABEL=DELETE
//      1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
//      [CNXLIB]
//
// EXEC IESVCLUP,SIZE=AUTO
//      1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
//      [CNXLIB]
//

```

Replace [?] [X]

Find what: [CNXLIB] [Find Next]

Replace with: CNXVRR [Replace]

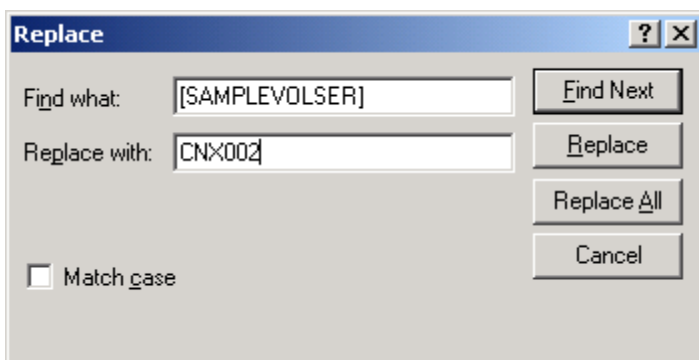
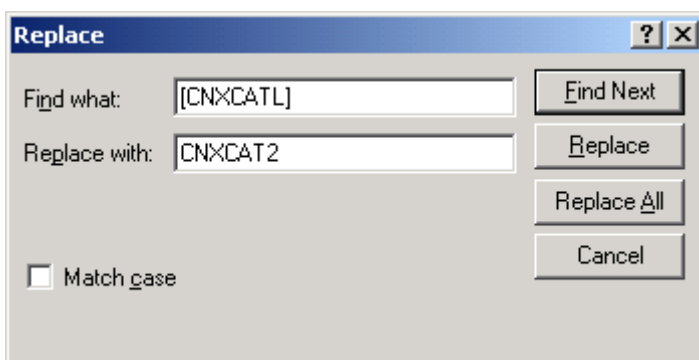
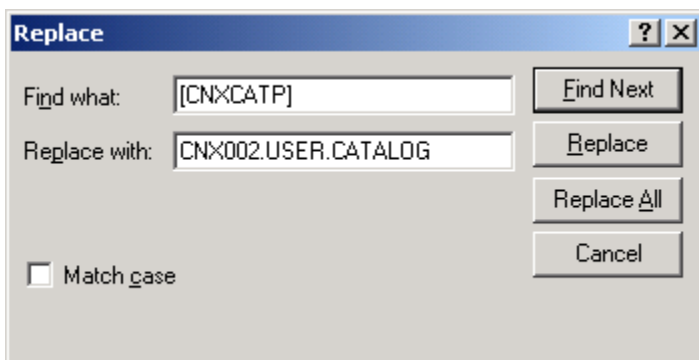
[Match case] [Replace All] [Cancel]

Replace [?] [X]

Find what: [CNXLBP] [Find Next]

Replace with: CONNX.LIBRARY [Replace]

[Match case] [Replace All] [Cancel]



3. After making the substitutions in the cnxlibv_.dat template, visually inspect the JCL and move each line continuation character (a rightmost capital X preceded by a comma and one or more blanks) to column 72; truncate all data beyond column 72.

The control records for the standard label delete and add job steps are column-sensitive; examples of unsubstituted and substituted control records follow:

Column Rule Lines

Delete Step, template version:


```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// DLBL [CNXLIB], '[CNXLIB]',,VSAM, X
CAT=[CNXCATL], DISP=(OLD, KEEP)

```

Substituted example:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// DLBL CNXV8R9, 'CONNX.LIBRARY',,VSAM, X
CAT=CNXCAT2, DISP=(OLD, KEEP)

```

Add Step, template version:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// EXEC IESVCLUP, SIZE=AUTO
A [CNXLIB] [CNXLIB] [CNXCATL] OLD KEEP

```

Substituted example:

```

1      2      3      4      5      6      7      8
123456789012345678901234567890123456789012345678901234567890
// EXEC IESVCLUP, SIZE=AUTO
A CONNX.LIBRARY CNXV8R9 CNXCAT2 OLD KEEP

```

4. Delete the column rule lines in the template and save the cnxlibv_.dat file as a new file; for example, cnxlibv0.jcl.
5. Via the Windows command prompt, start an ftp session to the target VSE host. Change directory to the VSE POWER Reader queue and submit the cnxlibv0.jcl job:


```

>ftp
>open 123.456.788.812
>user [USER]
>password [PASSWORD]
>cd POWER\RDR\A
>put cnxlibv0.jcl
      
```
6. This job should execute in less than a minute. Within the same ftp session, retrieve the output JCL from the POWER List Queue:


```

>cd "..\..\LST\A"
>dir
      
```
7. Given the default job name defined in the cnxlibv_.dat template, a job with the following naming convention is displayed:

- CNXLIB.00037.00 04/17/03 08:32 94 11 79 3 D SYSTCPIP
8. Retrieve the job output by issuing the following command:

```
>get CNXLIB.00037.00
```
 9. The sample output below is from a run which deletes CONNX sub-libraries and libraries in Adabas SQL Gateway-managed space before redefining new versions:

```
1// JOB  CNXLIB DEFINE CONNX FOR Adabas SQL Gateway/VSE Adabas SQL
      Gateway-MANAGED LIBRARY      DATE 04/17/2003, CLOCK 08/32/01
// EXEC LIBR,PARM='MSHP'

1      DELETE SUBLIB=CNXVRR.CNTL
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVRR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVRR.COPYBOOK
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVRR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVRR.OBJLIB
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVRR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVRR.RUNLIB
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVRR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
0      DELETE SUBLIB=CNXVRR.SOURCE
0L252I NO LABEL INFORMATION FOUND FOR LIBRARY CNXVRR
0L027I ABNORMAL END DURING DELETE COMMAND PROCESSING
0L113I RETURN CODE OF DELETE IS 8
11S55I  LAST RETURN CODE WAS 0008
// EXEC IDCAMS,SIZE=AUTO
1IDCAMS  SYSTEM SERVICES                                TIME: 08:32:02
      04/17/2003   PAGE   1

0
      DELETE CONNX..LIBRARY                               -
      CLUSTER PURGE                                     -
      CATALOG (CNX002.USER.CATALOG)

0IDC3012I ENTRY CONNX.LIBRARY NOT FOUND
IDC3009I ** Adabas SQL Gateway CATALOG RETURN CODE IS 8 - REASON CODE
IS IGG0CLCG-6
IDC0551I **ENTRY CONNX.LIBRARY NOT DELETED
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 8
11S55I  LAST RETURN CODE WAS 0008
// OPTION STDLABEL=DELETE
      CNXVRR
1L63I  LABEL CNXVRR NOT FOUND
// EXEC IESVCLUP,SIZE=AUTO
```

```

1// EXEC IDCAMS,SIZE=AUTO
1IDCAMS  SYSTEM SERVICES                                TIME: 08:32:05
        04/17/2003  PAGE    1
0
  DEFINE CLUSTER ( -
    NAME ( CONNX..LIBRARY )                -
    TRACKS (15 15 )                        -
    SHAREOPTIONS (3)                       -
    RECORDFORMAT (NOCIFORMAT)              -
    VOLUMES (CNX002)                       -
    NOREUSE                                -
    NONINDEXED                             -
    TO (99366))                            -
    DATA (NAME ( CONNX..LIBRARY.@D@ ) )    -
    CATALOG (CNX002.USER.CATALOG)
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
  IF LASTCC NE 0 THEN CANCEL JOB
0IDC0204I PRECEDING COMMAND BYPASSED DUE TO CONDITION CODES
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
11S55I  LAST RETURN CODE WAS 0000
  // OPTION STDLABEL=ADD
  // DLBL CNXVVR, 'CONNX..LIBRARY',,Adabas SQL
    Gateway,CAT=CNXCAT2,DISP=(OLD,KEEP)
  // EXEC IESVCLUP,SIZE=AUTO
1// EXEC LIBR,PARM='MSHP'
1      DEFINE LIB=CNXVVR                      REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.CNTL              REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.COPYBOOK          REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.OBJLIB            REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.RUNLIB            REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
0      DEFINE SUBLIB=CNXVVR.SOURCE            REPLACE=YES
0L113I RETURN CODE OF DEFINE IS 0
11S55I  LAST RETURN CODE WAS 0000
  EOJ CNXLIB      MAX.RETURN CODE=0008                      DATE
        04/17/2003, CLOCK 08/32/13,      DURATION 00/00/12

```

Notes:

For a first-time run, the return codes from the first step:

// EXEC LIBR,PARM='MSHP'

is 8, which is expected.

10. For a first-time run, the return code from the IDCAMS statement:

DELETE CONNX.VXXRXX.LIBRARY -

CLUSTER PURGE -

CATALOG (CNX001.USER.CATALOG)

will be 8, which is expected. All subsequent return codes should be 0 or 4.

11. The VSE volume and Adabas SQL Gateway user catalog physical data set and data label names in the above example are CNX002, CNX002.USER.CATALOG, and CNXCAT1. These values are substituted into template variables [SAMPLEVOLSER], [CNXCATP], and [CNXCATL].
12. Make the CONNX for Adabas SQL Gateway - VSE library available for FTP access.
13. Depending on the TCP/IP security rules of the target VSE system, the newly allocated CONNX library may not be immediately available for ftp read/write access. The Adabas SQL Gateway library can be manually defined to the VSE TCP/IP partition via VSE console commands as follows:

msg f7

AR 0015 1140I READY

F7-0100 IPN300I Enter TCP/IP Command

102 def file,pub=CNXVVRR,dlbl=CNXVVRR,type=lib,READONLY=NO

F7 0098 IPN264I File defined, Dataset: CNXVVRR

The CONNX for VSE-Adabas SQL Gateway library can be permanently defined to the TCP/IP partition via the batch JCL provided in member IPINIT00 of the TCP/IP for VSE installation library. The control record syntax is identical to the console command:

DEFINE FILE,PUBLIC='CNXVVRR',DLBL=CNXV8R8,TYPE=LIBRARY,READONLY=NO

To verify that the new CNXVVRR library is defined and available for ftp, start a Command Prompt window, and then an ftp session to the target host. Once logged on, you can navigate to the CNXVVRR directory and list the subdirectories:

230 User logged in, proceed.

ftp> cd cnxv8r9

250 Requested file action okay, completed.

ftp> dir

200 Command okay.

150 File status okay; about to open data connection

CNTL <Sub Library> 0 1 04/17/03 08:32

COPYBOOK <Sub Library> 0 1 04/17/03 08:32

OBJLIB <Sub Library> 0 1 04/17/03 08:32

RUNLIB <Sub Library> 0 1 04/17/03 08:32

SOURCE <Sub Library> 0 1 04/17/03 08:32

14. You are ready to begin the Adabas SQL Gateway/VSE automated installation. From the **Start** menu, select **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. Follow the steps described in the next section, "Part 2: Installing the Adabas SQL Gateway Components on a VSE Server".

Part 2: Installing the Adabas SQL Gateway Components on a VSE system

General Information about the Adabas SQL Gateway Server

This step copies the Adabas SQL Gateway server components from a client PC to a VSE target. The CONNX Administrator component must be installed on the client PC.

About the Adabas SQL Gateway - VSE User Account and Password

When installing the Adabas SQL Gateway components, select a VSE user account name and password with privileges that allow you to create VSE libraries in VSAM-managed space and to copy, read, and write files via FTP to the target host.

To install the Adabas SQL Gateway - VSE components

Before attempting to install the Adabas SQL Gateway for VSE, create a Software AG CICS High Performance Stub Routine. For information on how to create this routine, see the Software AG "Adabas Installation Manual (VSE/ESA)".

1. Select a computer on which the CONNX Server Installation component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The **CONNX Server Component Installation** dialog box appears. Select the **VSE** tab.

The screenshot shows the 'CONNX Server Component Installation' dialog box with the 'VSE' tab selected. The dialog box has a title bar with standard window controls and a 'Files' menu. Below the title bar, there's a subtitle 'CONNX Server Component Installation' and a prompt 'Enter the platform type, transfer method and login credentials.' To the right of the prompt is a red puzzle piece icon. The main area is divided into several sections: 'CONNX Server Library' with fields for 'CONNX Library DSN' (containing 'CONNX'), 'VOLSER', 'CONNX Library Label' (containing 'CNXVXRX'), 'Catalog DSN', and 'Catalog Label'; 'Load Library Parameters' with 'Target RDO File' (containing 'CICS.CSD') and 'ADABAS Parameters' with 'ADA Link Stub Lib' (containing 'SAG.ADA.LOAD'); 'Login Information' with fields for 'Server', 'Username', and 'Password'; 'Product' with radio buttons for 'ADABAS' (checked) and 'VSAM'; 'TCP/IP Stack' with radio buttons for 'CSI 1.5E and below', 'CSI 1.5E and above' (checked), and 'Barnard'; and 'Mode' with radio buttons for 'Active' and 'Passive' (checked). At the bottom right, there's a text field containing 'PRD2.TCPIP'. At the very bottom are 'Install' and 'Done' buttons.

2. In the **CONNX Server Library** area, enter the fully qualified VSE cluster name which contains the CONNX VVRR Adabas SQL Gateway library in **CONNX Library DSN** and its VOLSER in the **VOLSER** text box. Enter the CONNX Adabas SQL Gateway library name defined in the CONNX library cluster in **CONNX Library Label**, the data set name of the VSE user catalog where the CONNX Adabas SQL Gateway sample files and managed library are defined in **Catalog DSN**, and the system data label which references the physical data set name of the Adabas SQL Gateway User Catalog in **Catalog Label**.

Note: If **CONNX Library Label** is blank, the library name defaults to CNXWRR. If **Catalog DSN** is blank, CONNX files are defined in the default VSE system catalog. If **Catalog Label** is blank, the data label defaults to IJSYSCT.

3. In the **Login Information** area, type the TCP/IP symbolic host name or dotted numeric address for your VSE server in the **Server** text box, a privileged user account name in the **User ID** text box and

its password in the **Password** text boxes. Under **Product**, click **VSAM** and under **Server Type**, click **CICS**.

4. In the **TCP/IP Stack** area, if you are using the Barnard TCP/IP stack rather than CSI, select the **Barnard** radio button. For **Mode**, select either **Active** or **Passive** FTP mode. Unless your system is not configured to support Passive mode FTP, you should select Passive. If the FTP transfer fails for any reason other than the user id or password was entered incorrectly, switch modes and try again.

Enter the TCP/IP stack location in **TCP/IP HLQ**.

Caution: To enable the environment variable for the Barnard TCP/IP stack, set [CNXBARNARD](#) to 1 in the CONNX Configuration Utility.

5. In the **CICS Parameters** area, see the table in [Required and Optional Entry Fields for VSAM TCP/IP Server Setup](#) for a detailed explanation of the information required in **Target RDO File** and **Adabas Link Stub Lib**.

6. In the **VSAM Samples Group** area, if you clear the **Create** check box, CONNX will not create the CONNX VSAM sample KSDS, RRDS, and ESDS data sets. If the **Create** check box is selected, the CONNX VSAM sample KSDS, RRDS, and ESDS data sets and the VSAM clusters described in [VSAM Objects for CONNX.CICS.SAMPLES](#) are created, defined to the target CICS region, and given the data set prefix specified in **Database HLQ**.

Note: The default dataset prefix for **Database HLQ** is CONNX.SAMPLES.

7. Click the **Install** button. This action starts an FTP session and copies the CONNX components from the client PC to the target host. A message confirming that the FTP session has started appears in the status bar..

8. Allow several minutes for the file transfer process to complete.

9. Click the **Close** button. The **CONNX Server Component Installation** dialog box closes.

Required and Optional Entry Fields for Adabas TCP/IP Server Setup - VSE

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS UNIX/Linux

CONNX Server Library

CONNX Library DSN: CONNX

VOLSER:

CONNX Library Label: CNXVXRX

Catalog DSN:

Catalog Label:

Load Library Parameters

Target RDO File: CICS.CSD

ADABAS Parameters

ADA Link Stub Lib: SAG.ADA.LOAD

Login Information

Server:

Username:

Password:

Product

☒ ADABAS

☐ VSAM

TCP/IP Stack

☐ CSI 1.5E and below

☒ CSI 1.5E and above

☐ Barnard

Mode

☐ Active

☒ Passive

PRD2.TCPIP

Install Done

Group Box	Term	Description
Login Information	Server	A symbolic or dotted numeric TCP/IP address, for example: MVS or 123.123.123.123
	User ID	Your 1-8 character VSE logon ID. All characters convert from lowercase to uppercase.
	Password	Your 1-8 character VSE password. All characters convert from lowercase to uppercase.
TCP/IP Stack	CSI	CSI International TCP/IP connectivity.
	Barnard	The Barnard TCP/IP communications stack.
	TCP/IP HLQ	The TCP/IP symbolic host name or dotted numeric address for your VSE system.

CONNX Server Library	CONNX Library DSN	A one- or multi-part high-level data set name, which is used to create the CONNX installation sequential and partitioned data sets on the target host. In the current example, the library DSN is CONNX.CICS.
	VOLSER	The DASD volume serial on which the CONNX.Adabas sequential and partitioned data sets are created. Optional: The DASD volume serial on which the CONNX.ADABAS sequential and partitioned data sets are created.
	CONNX Library Label	The 1-44 character fully qualified VSE Adabas SQL Cluster name that contains the CONNX VVRR Adabas SQL Gateway library. The default is CNXV9R0.
	Catalog DSN	The 1-44 character physical data set name of the VSE Adabas SQL Gateway user catalog into which the Adabas SQL Gateway-managed library are defined. If left blank, the CONNX files are defined in the default VSE system catalog.
	Catalog Label	The 1-7 character system data label which references the physical data set name of the Adabas SQL Gateway User Catalog.
CICS Parameters	Target RDO File	System Management File location.
Adabas Parameters	Link Stub Library	Location of the Software AG CICS High Performance Stub Routine. (See page 75 in the Software AG "Adabas Installation Manual VSE/ESA").

Part 3: Installing the CONNX CICS components

To install the selected components to the VSE target

For CICS/VSE 2.3 and above installations:

- The initial FTP step creates an installation JCL member on the target VSE system in the Adabas library. Given the default physical Adabas file name of CONNX.VVRR.LIBRARY and the default library.sublib name of CNXVVRR.CNTL, submit the JCL by starting an interactive DITTO session, editing member CNXINST0.JCL to conform to your installation standards, and submitting the revised JCL to the VSE reader queue.
- Save the changes to CNXINST0.JCL with the DITTO Save command, then submit the JCL to the VSE POWER reader queue with the DITTO submit command.

Summary of the CONNX for Adabas-VSE Installation Procedure

For CICS/VSE 2.4 and above installations:

Steps 1-4: Execute the linkage editor against object code for the following CONNX programs:

CNXCFCG	Configuration + TCP/IP Listener start/stop program
CNXRUN	TCP/IP Listener
CNXADD0	TCP/IP Server
CNXSTOP	Automated shutdown program for CNXRUN

The four link-edit steps must end in return codes <= 4.

Step 5: Executes the IBM batch CICS RDO update utility program DFHCSDUP to define CONNX files, transactions, and programs to the target CICS partition RDO cluster. The first execution returns warnings for the unconditional DELETES of CONNX objects from the undefined RDO Group CNXVVR. Such warnings can be safely ignored. After all CONNX RDO objects are defined, a list of the objects created in RDO group is generated.

Step 6: Executes the IBM Access Methods Services utility program IDCAMS to DELETE/DEFINE the configuration file (CNXCFCG) and the sample KSDS and RRDS files.

Step 7: Executes the IBM Access Methods Services utility program IDCAMS to copy (REPRO) the sample data from the temporary ESDS files created by the initial FTP step to the corresponding KSDS and RRDS files. After the REPRO steps, the temporary ESDS files are deleted, reallocated, and loaded with data from the corresponding KSDS files. This step saves space. Finally, two alternate indexes each for the sample KSDS equipment and order files are defined and built. The return code from this step should be <= 4.

Step 8: To submit the CONNX for VSE batch installation job to the VSE POWER reader queue:

1. Change the job name, class, and PDEST/LDEST keywords to match installation standards.
2. For this example, all CONNX for VSE modules are defined in the CNXVVR library which resides in Adabas dataset CONNX.LIBRARY in user catalog CNXCAT2.
3. The // LIBDEF OBJ search list points to the default lib.sublib for the CONNX for VSE/Adabas object code (CNXVVR.OBJLIB); the PRD2.TCPIP library is the default target for the TCP/IP for VSE modules from Connectivity Systems, Inc. Change these entries as necessary to match your VSE installation standards.

Save the changes to CNXINST0.JCL; then enter submit on the DITTO command line to copy the JCL to the VSE POWER reader queue.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CNXINST0.JCL      Library CNXVVRR.CNTL      Col 1      Format CHAR
                                SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 * $$ JOB JNM=CNXINST,PDEST=(*,CNX),LDEST=(*,CNX),CLASS=0
00002 // JOB      CNXINST
00003 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM,                                X
00004           CAT=CNXCAT2,DISP=(OLD,KEEP)
00005 // LIBDEF  OBJ,SEARCH=(CNXVVRR.OBJLIB,PRD2.TCPIP)
00006 // LIBDEF  PHASE,CATALOG=CNXVVRR.RUNLIB
00007 // OPTION  CATAL
00008 ACTION  MAP
00009 PHASE  CNXCFG,*
00010 INCLUDE DFHEAI
00011 INCLUDE DFHEAIO
00012 INCLUDE CNXCFG

Command ==> submit      Scroll CSR
F1=Help  F2=SplitJoin  F3=Save+Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd
F8=Fwd  F10=RChange  F11=CRetrieve  F12=Quit

MA  a  22/021

```

Completing CICS Customization - Adabas

Once the Adabas SQL Gateway - VSE CICS components are built, they must be defined to the target CICS partition. This process includes updating the CICS DCT (Destination Control Table), and adding the CONNX library to the CICS LIBDEF SEARCH list.

To add SAS/C transient data queue definitions to the CICS Startup JCL

The CICS target partition startup JCL must be modified to specify the new DLBLs/datasets defined as SAS/C 7.00 Transient Data Queues, for example:

```
// ASSGN SYS012,DISK,VOL=volser,SHR
// DLBL STDOUT,'CONNX.CICS.STDOUT',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
// DLBL STDERR,'CONNX.CICS.STDERR',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
// DLBL STGRPT,'CONNX.CICS.STGRPT',0,SD
// EXTENT SYS012,volser,1,0,aaa,bbb
```

Notes:

1. *Substitute a valid VSE DASD volume for volser.*
2. *Substitute a valid track start address and the number of tracks or blocks for each instance of aaa,bbb above. The recommended allocation for STDOUT is at least 15 tracks; the STDERR and STGRPT data sets can be initially allocated with 5 tracks each.*
3. *The data set prefix CONNX.CICS is suggested but not required; change it as appropriate to conform to your installation standards.*

Installation Verification of CONNX CICS Components

Option 1 - RDO: If you request to install the CONNX CICS components, the installation job executes the IBM DFHCSDUP utility to define the CONNX programs, transactions, and Adabas configuration and sample files to the target CICS partition RDO Adabas file. You can verify the installation of the CONNX CICS components by logging on to the target CICS partition and issuing the following command:

```
CEDA EXPAND GR(CNX)
```

For CICS 2.4 and above installations, the following components are defined:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
EXPAND GR(CNX0809)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
CNXCFG    FILE      CNXVVR     03.106    07.51.24
CNXCSTE    FILE      CNXVVR     03.106    07.51.35
CNXCSTK    FILE      CNXVVR     03.106    07.51.34
CNXCSTR    FILE      CNXVVR     03.106    07.51.35
CNXEQE     FILE      CNXVVR     03.106    07.51.51
CNXEQI1    FILE      CNXVVR     03.106    07.51.53
CNXEQI2    FILE      CNXVVR     03.106    07.51.54
CNXEQK     FILE      CNXVVR     03.106    07.51.50
CNXEQP1    FILE      CNXVVR     03.106    07.51.55
CNXEQP2    FILE      CNXVVR     03.106    07.51.56
CNXEQR     FILE      CNXVVR     03.106    07.51.52
CNXORE     FILE      CNXVVR     03.106    07.51.47
CNXORI1    FILE      CNXVVR     03.106    07.51.57
CNXORI2    FILE      CNXVVR     03.106    07.51.58
CNXORK     FILE      CNXVVR     03.106    07.51.45
CNXORP1    FILE      CNXVVR     03.106    07.51.59
+ CNXORP2  FILE      CNXVVR     03.106    07.52.01

                                SYSID=CIC2  APPLID=PRODCICS
RESULTS: 1 TO 17 OF 31          TIME: 16.44.09  DATE: 03.107
PF 1 HELP      3 END 4 TOP 5 BOT 6 CSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
MA b                                                  24/002

```

The group name suffix "VVR" represents version and release numbers, which can change with every release.

List of Installed Components on the CICS/VSE Partition

After proper installation, the following components appear in the CNXVVRR library in a VSAM-managed space.

CNXVVRR.CNTL

Member	Description
@REMOVE0.JCL	CONNX for VSAM/VSE Complete Uninstall
@REMOV10.JCL	CONNX for VSAM/VSE Partial Uninstall
CNXINST0.JCL	CONNX for VSAM/VSE Installation
CNXINS10.JCL	CONNX for VSAM/VSE JCL for CICS V2R3 and below partitions

CNXVVRR.OBJLIB

Member	Description
CNXCFCG.OBJ	CONNX CICS/VSE configuration program object code
CNXRUN.OBJ	CONNX CICS/VSE TCP/IP listener program object code
CNXSTOP.OBJ	CONNX CICS/VSE Automated TCP/IP Listener shutdown program object code
CNXADD0.OBJ	CONNX CICS/VSE TCP/IP server program object code
CNXSIGN.OBJ	CONNX CICS/VSE 2.3 External Security Interface program object code
VSOCKET.OBJ	CONNX CICS/VSE TCP/IP object code stub for socket() function

CNXVVRR.RUNLIB

Member	Description
CNXCFCG.PHASE	CONNX CICS/VSE configuration program PHASE
CNXRUN.PHASE	CONNX CICS/VSE TCP/IP listener program PHASE
CNXSTOP.PHASE	CONNX CICS/VSE Automated TCP/IP listener shutdown program PHASE
CNXADD0.PHASE	CONNX CICS/VSE TCP/IP server program PHASE

CNXVVRR.SOURCE

Member	Description
CNXFCT0.ASM	File Control Table source for CONNX Configuration and Sample VSAM Files for CICS/VSE 2.4 and above

CNXPPT0.ASM	Program Processing Table source for CONNX programs
SASDCT.ASM	SAS/C Destination Control Table source
SASPPT.ASM	SAS/C Program Processing Table source

Part 4: CONNX Configuration Utility

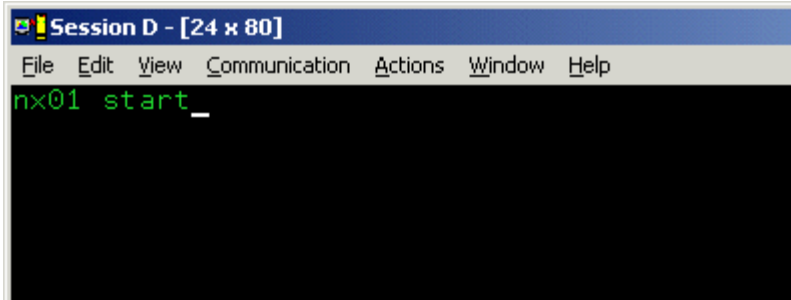
Configuration Utility

The CONNX configuration utility for Adabas/CICS enables the user to start the TCP/IP listener, and to set, remove, and display current CONNX variables.

To start the CONNX CICS/Adabas TCP/IP Listener

The listener process waits for and verifies connections with client machines.

1. After logging on to the CICS region, enter **NX01 Start**, and then press **<Enter>**.

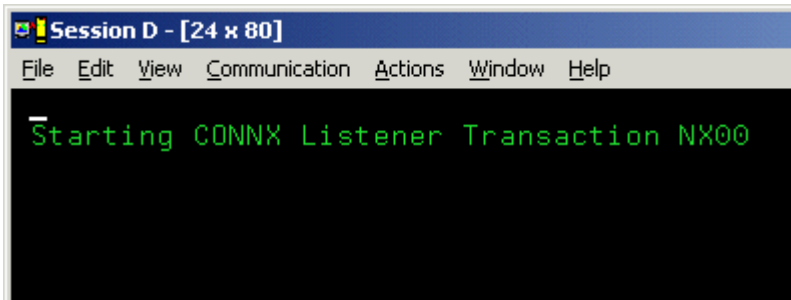


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
nx01 start_

```

2. The CONNX TCP/IP Listener starts.

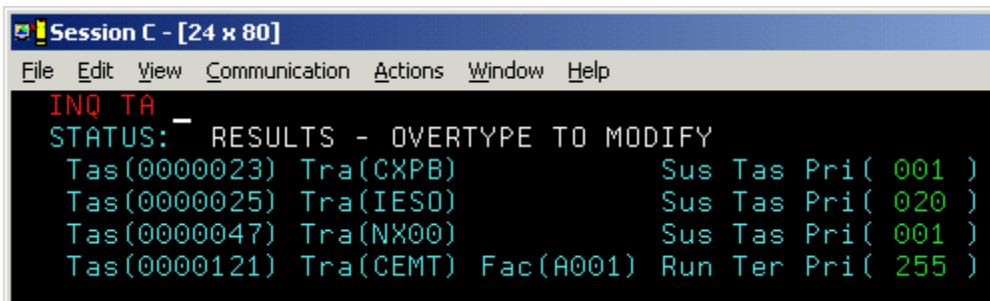


```

Session D - [24 x 80]
File Edit View Communication Actions Window Help
Starting CONNX Listener Transaction NX00

```

3. Verify start by typing **CEMT INQ TA**. Press **<Enter>**. The screen displays a list of transactions currently running. Verify that **Tas(0000xxx) Tra(NX00)** is listed as one of the transactions.



```

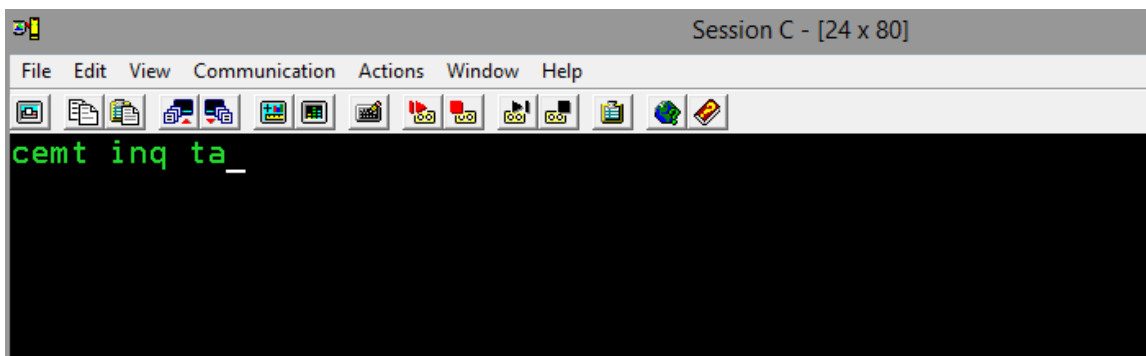
Session C - [24 x 80]
File Edit View Communication Actions Window Help
INQ TA
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000023) Tra(CXPB) Sus Tas Pri( 001 )
Tas(0000025) Tra(IES0) Sus Tas Pri( 020 )
Tas(0000047) Tra(NX00) Sus Tas Pri( 001 )
Tas(0000121) Tra(CEMT) Fac(A001) Run Ter Pri( 255 )

```

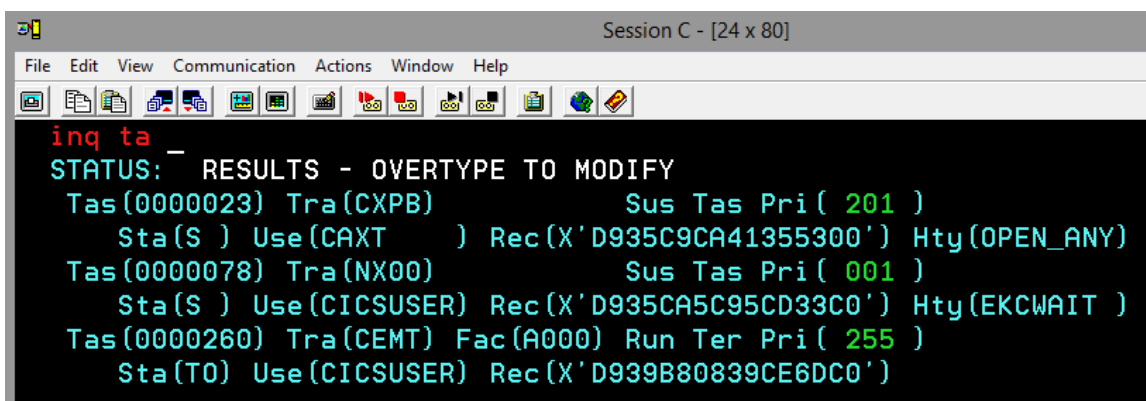
To stop the CONNX CICS/Adabas TCP/IP Listener

You can stop the CONNX CICS/Adabas TCP/IP listener program with the NX01 transaction from an authorized 3270 terminal session. The NX01 STOP command gracefully terminates the CONNX CICS/Adabas TCP/IP listener, and permits it to be restarted on the same or on a different TCP/IP port.

1. Verify that the CONNX CICS TCP/IP listener transaction/program (NX00/CNXRUN) is running by typing **cemt inq ta**. Press **<Enter>**.



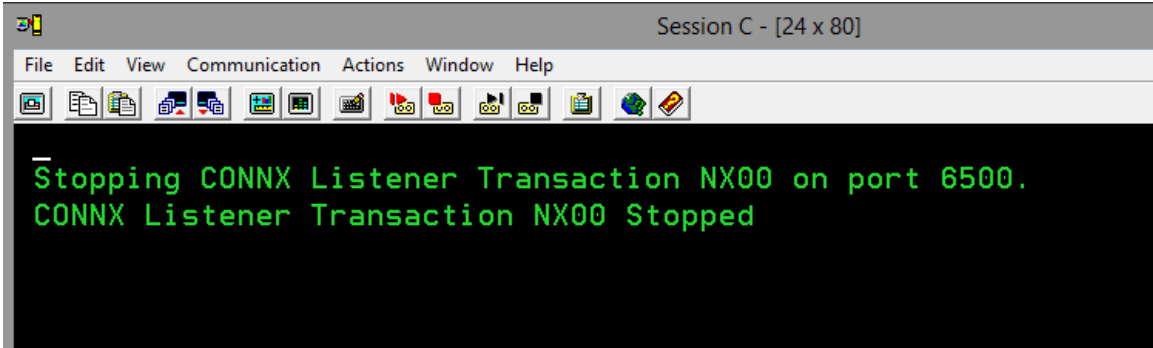
2. Verification appears. Press the F3 key, and then press **<Clear>**.



3. Stop the CONNX CICS TCP/IP listener by typing **nx01 stop**, and then press **<Enter>**.



4. Verification appears.

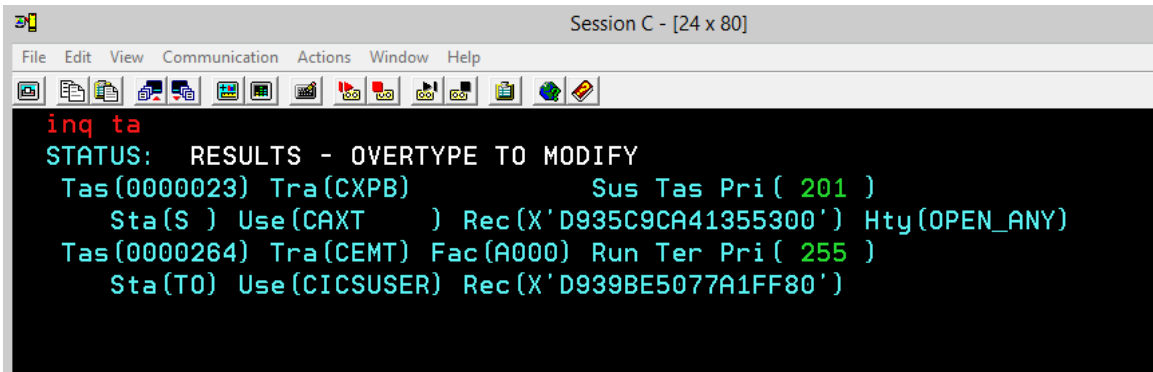


```

File Edit View Communication Actions Window Help
Stopping CONNX Listener Transaction NX00 on port 6500.
CONNX Listener Transaction NX00 Stopped

```

5. You can verify further that the CONNX CICS TCP/IP listener transaction/program (NX00/CNXRUN) is stopped by issuing a second **CEMT INQ TA** command, which shows that the NX00 task has been stopped. You can restart the CONNX CICS/Adabas TCP/IP Listener program on the same TCP/IP port, or you can change the TCP/IP port using the NX01 CNXRUNPORT command, and restart the listener on a different port.



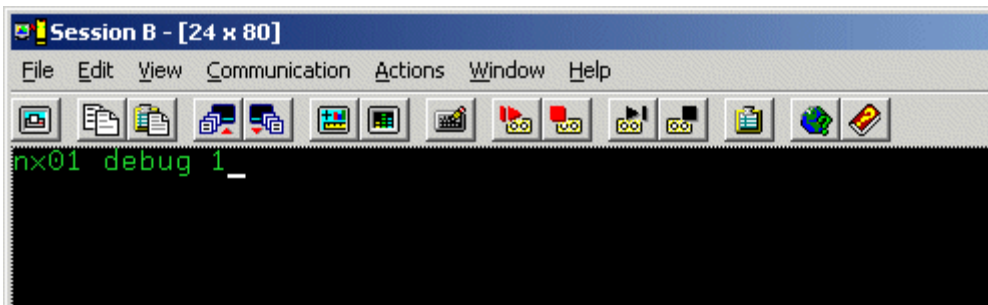
```

File Edit View Communication Actions Window Help
inq ta
STATUS: RESULTS - OVERTYPE TO MODIFY
  Tas(0000023) Tra(CXPB)          Sus Tas Pri( 201 )
    Sta(S ) Use(CAXT      ) Rec(X'D935C9CA41355300') Hty(OPEN_ANY)
  Tas(0000264) Tra(CEMT) Fac(A000) Run Ter Pri( 255 )
    Sta(T0) Use(CICSUSER) Rec(X'D939BE5077A1FF80')

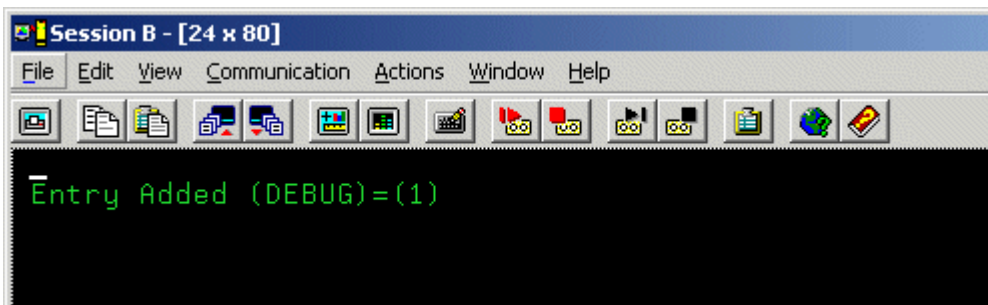
```

To set a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE> <VALUE>**, and then press **<Enter>** to turn on the CONNX variable function. In this example, the DEBUG variable is used.

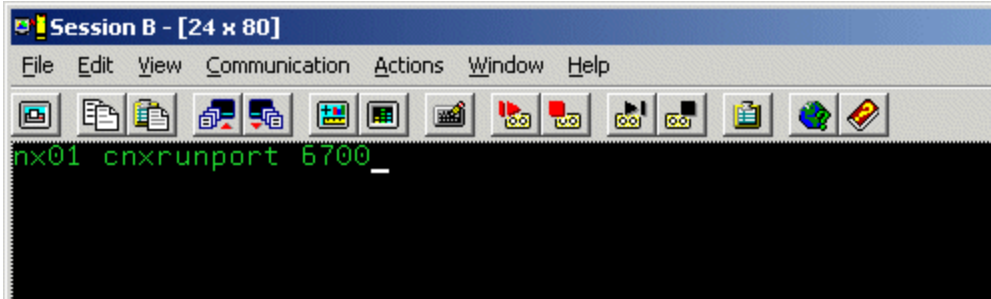


2. The variable is set, and verification appears.

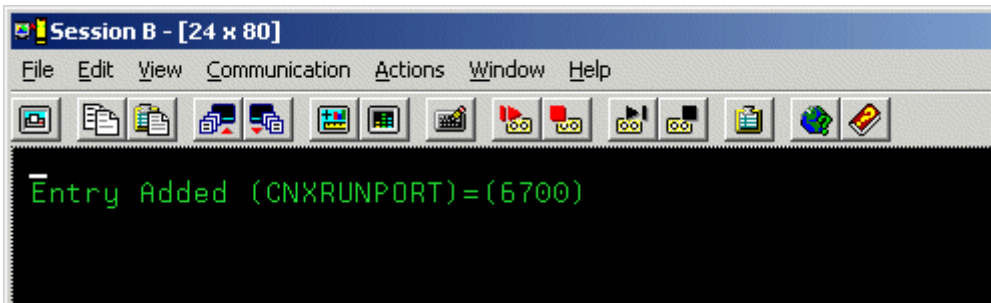


To change the number of the CONNX TCP/IP port

1. Clear the screen, type **NX01 CNXRUNPORT <NewPortNumber>**, and then press **<Enter>**. (The default CONNX port is 6500.) The new port number used in this example is 6700.
2. If you change this port, all clients must be updated to the same port number. See TCP/IP Connection Port in the *CONNX User Reference Guide*, available online, on the CONNX CD-ROM, and within the CONNX product, for more information on changing the TCP/IP connection port for client machines.

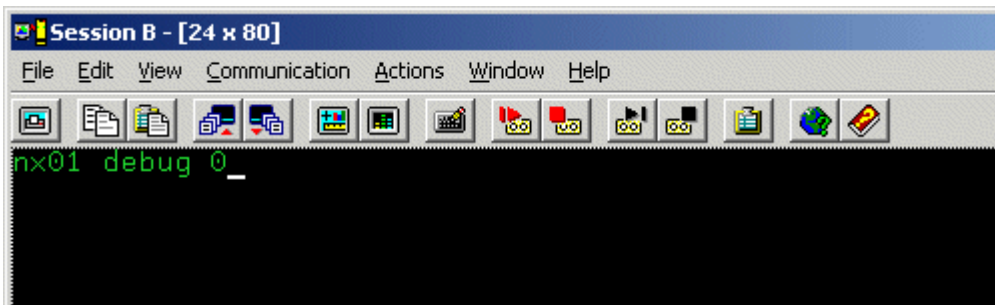


3. The port number is changed, and verification appears.

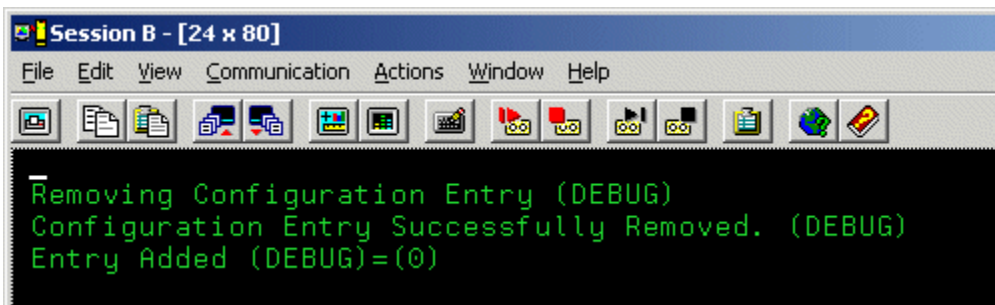


To change a CONNX environment variable

1. Clear the screen, type **NX01 <VARIABLE>**, and then press **<Enter>**. In this example, in which the variable was set in [To set a CONNX environment variable](#), the **DEBUG** variable is used.

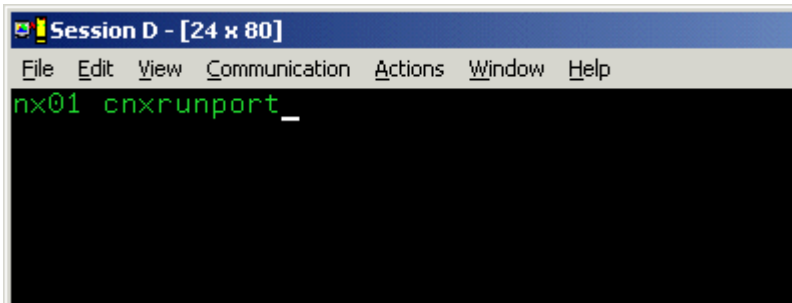


2. The variable is removed, and verification appears.

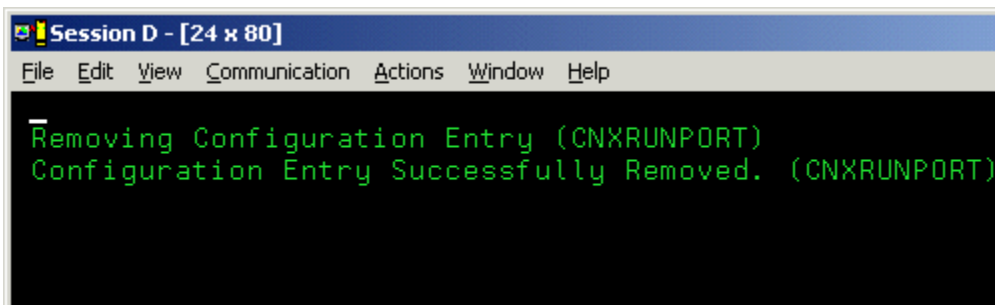


To convert the CONNX port number to the default

1. Clear the screen, type **NX01 CNXRUNPORT**, and then press **<Enter>**.



2. The current entry for CNXRUNPORT is removed, and verification appears. Removal of the CNXRUNPORT variable restores the default value of 6500.

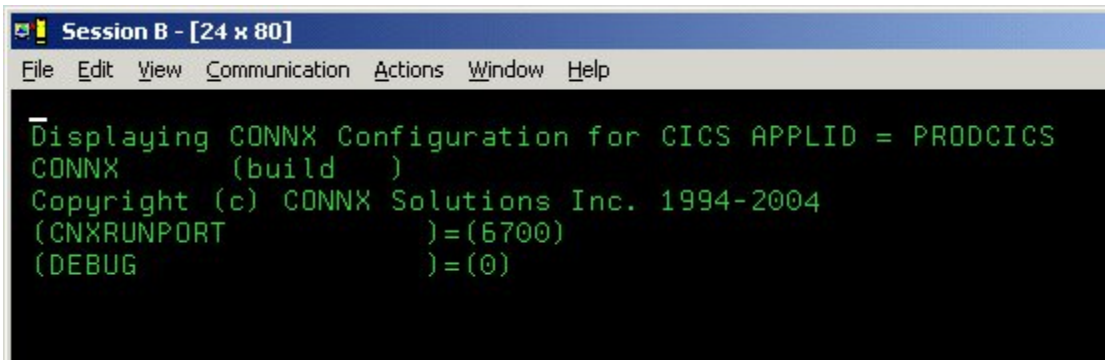


To enable the Barnard TCP/IP Stack

1. Clear the screen, type **NX01 cnxbarnard 1**, and then press **<Enter>** to turn on the CONNX variable function that enables the Barnard TCP/IP stack.
2. The variable is set, and verification appears.

To display the current environment variable(s)

1. Type **NX01**, and then press **<Enter>** to display the current configuration.



```
Session B - [24 x 80]
File Edit View Communication Actions Window Help

Displaying CONNX Configuration for CICS APPLID = PRODCICS
CONNX      (build      )
Copyright (c) CONNX Solutions Inc. 1994-2004
(CNXRUNPORT      )=(6700)
(DEBUG           )=(0)
```

2. To test your connection, start the CONNX Data Dictionary Manager application and follow the import procedures described in the sections on the Adabas SQL Gateway found in the CONNX User Reference Guide available online, on the CONNX CD-ROM, and within the CONNX product.

Part 5: Adabas SQL Gateway for VSE: Uninstallation Procedures

Uninstallation Procedures

The Adabas SQL Gateway TCP/IP Server Setup client application transfers the CONNX components to the target VSE host via FTP. After the initial FTP step, the CONNX installation continues via a batch job submission using VSE Ditto. The final installation steps on VSE platforms consist of manual updates to the CICS target partition Destination Control Table and the startup JCL.

The Adabas SQL Gateway provides batch uninstallation JCL for VSE platforms.

To uninstall Adabas SQL Gateway - VSE

Given a default CONNX installation library name of CNXVVRR, the CONNX installation JCL library.sublib is CNXVVRR.CNTL. Member @REMOVE0.JCL contains the batch uninstallation JCL.

1. Log on to the target CICS partition and stop the CONNX NX00 TCP/IP listener transaction NX00 by entering:
NX01 STOP
2. Next, shut down CICS by entering:
CEMT PERF SHUT IMMED
3. After the CICS partition shuts down, you can submit the JCL in member @REMOVE0.JCL via the CICS Ditto transaction. Note that the first control record of the // EXEC LIBR step deletes the CNXVVRR.CNTL library.sublib, so be sure to submit the job and deallocate the CONNX library by exiting from the CICS Ditto transaction.

Example:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit
Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 * $$ JOB JNM=CNXREMOV,PDEST=(*,CNX),LDEST=(*,CNX),CLASS=0
00002 // JOB      CNXREMOV
00003 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM,
00004           CAT=CNXCAT2,DISP=(OLD,KEEP)
00005 // DLBL VSESPUC,'VSESP.USER.CATALOG',,VSAM
00006 // DLBL DFHCSD,'CICS.CSD',,VSAM,CAT=VSESPUC
00007 // EXEC DFHCSDUP,SIZE=DFHCSDUP,PARM='CSD(READWRITE)'
00008 DELETE GROUP(CNX0809)
00009 /*
00010 // DLBL IJSYSUC,'CNX002.USER.CATALOG',,VSAM
00011 // EXEC IDCAMS,SIZE=AUTO
00012 DELETE CONNX.V08R09.LIBRARY.CNXCFG -

Command ==>
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```

4. The IDCAMS DELETE step should return zeroes for all CONNX data sets which exist and are not allocated to another VSE process or job. The EXEC LIBR step to delete the Adabas-managed library and sublibraries may return an informational message to the effect that the deletes should be performed with Access Method Services. The CICS RDO DFHCSDUP utility step should return zeroes for deletion of the CNXVVRR group. The final IDCAMS DELETE step removes the cluster that contains the CNXVVRR library. The last IESVCLUP step removes the standard label for the CNXVVRR library, if one was created during installation.

Manual Steps: Adabas SQL Gateway - VSE Uninstallation

The startup JCL for the CICS partition on which CONNX has been installed has been manually updated to concatenate the CONNX application library.sublib to the LIBDEF search list. The references to CNXVRRR must be deleted.

Example:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CICS2.Z             Library IJSYSRS.SYSLIB      Col 1          Format CHAR
                                  SYSIPT data NO
1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00000 ***** Top of data *****
00001 $$$$ JOB JNM=CICS2,DISP=L,CLASS=8,EQJMSG=YES
00002 $$$$ LST CLASS=A,DISP=D,RBS=100
00003 // JOB CICS2          STARTUP OF SECOND CICS WITHOUT ICCF
00004 // OPTION SADUMP=5
00005 // OPTION SYSDUMPC
00006 // DLBL CNXV8R9,'CONNX.LIBRARY',,VSAM,CAT=CNXCAT2
00007 // DLBL CNXCICS,'CONNX.CICS.LIBRARY',,VSAM,CAT=CNXCAT2
00008 // LIBDEF *,SEARCH=(CNXCICS.CICS,                                X
00009                      CNXV8R9.RUNLIB,                              X
00010                      PRD2.SERV140E,                                X
00011                      PRD2.TCPIP,                                    X
00012                      PRD2.CONFIG,PRD1.BASED,PRD1.BASE,PRD2.PROD,    X

Command ==>                               Scroll CSR
F1=Help  F2=SplitJoin F3=Save+Exit F4=Left  F5=Right  F6=RFind  F7=Bkwd
F8=Fwd   F10=RChange F11=CRetrieve F12=Quit

MA  b                                     22/015

```

Additionally, three SAS/C-specific DLBLs (STDERR, STDOUT, and STGRPT) reference sequential datasets dedicated to the CICS extra-partition Temporary Data Queues SASE, SASO, and SASR. These data set references should be removed from the CICS Startup JCL:

Example:

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member CICS2.Z             Library IJSYSRS.SYSLIB      Col 1          Format CHAR
                                     SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00026 NOLOG
00027 // EXEC PROC=DTRCICS2                LABELS FOR CICS FILES
00028 // ASSGN SYS012,DISK,VOL=CNX002,SHR
00029 // DLBL STDOUT,'CONNX.SASC700.PRODCICS.STDOUT',0,SD
00030 // EXTENT SYS012,CNX002,1,0,12993,75
00031 // DLBL STDERR,'CONNX.SASC700.PRODCICS.STDERR',0,SD
00032 // EXTENT SYS012,CNX002,1,0,13068,5
00033 // DLBL STGRPT,'CONNX.SASC700.PRODCICS.STGRPT',0,SD
00034 // EXTENT SYS012,CNX002,1,0,13073,15
00035 *   WAITING FOR VTAM TO COME UP
00036 // EXEC IESWAITT
00037 $$/*
00038 // IF XMODEF8 = COLD THEN

Command ==>
F1=Help  F2=SplitJoin  F3=Save+Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd
F8=Fwd   F10=RChange  F11=CRetrieve  F12=Quit

MA  b                                     22/015

```

Finally, you can delete the sequential SAS TDQ datasets via the CICS Ditto Utility.

Running Adabas SQL Gateway - VSE Uninstallation Without Shutting Down CICS

You can perform a partial uninstallation of CONNX without shutting down the target CICS partition as follows:

1. Log on to the target CICS partition and stop the CONNX CICS TCP/IP Listener transaction NX00 by entering:
NX01 STOP
2. Next, use the CEMT transaction to close and disable the CONNX VSAM configuration and sample files.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
INQ FI(CNX*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Fil(CNXCFG ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXCSTE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXCSTK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXCSTR ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXEQI1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQI2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQP1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQP2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXEQR ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORE ) Vsa Ope Ena Rea Upd Add Bro
Fil(CNXORI1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORI2 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORK ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORP1 ) Vsa Ope Ena Rea Upd Add Bro Del
Fil(CNXORP2 ) Vsa Ope Ena Rea Upd Add Bro Del
+ Fil(CNXORR ) Vsa Ope Ena Rea Upd Add Bro Del

RESPONSE: NORMAL
TIME: 14.08.31 DATE: 04.18.03
PF 1 HELP 3 END 7 SBH 8 SFH 9 MSG 10 SB 11 SF
MA b 01/015
  
```

3. Edit the @REMOVE0 JCL member and delete the final three steps beginning with:
// EXEC LIBR,PARM='MSHP'


```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00091 // EXEC LIBR,PARM='MSHP'
00092      DELETE SUBLIB=CNXV8R9.CICS
00093      DELETE SUBLIB=CNXV8R9.CNTL
00094      DELETE SUBLIB=CNXV8R9.COPYBOOK
00095      DELETE SUBLIB=CNXV8R9.OBJLIB
00096      DELETE SUBLIB=CNXV8R9.RUNLIB
00097      DELETE SUBLIB=CNXV8R9.SOURCE
00098 /*
00099 // EXEC IDCAMS,SIZE=AUTO
00100      DELETE CONNX.LIBRARY
00101      CLUSTER PURGE
00102      CATALOG (CNX002.USER.CATALOG)
00103 /*

Command ==>
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```

4. Save and submit the modified JCL via the CICS Ditto transaction.
5. After this job completes and the target CICS partition is shut down, start another CICS Ditto session and submit JCL member @REMOV10 to delete the CNXV8R8 library and the standard label. Follow the manual steps above to complete the CONNX uninstallation.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help
Process View Options Help

DITTO/ESA for VSE          LE - Library Member Edit

Member @REMOVE0.JCL      Library CNXV8R9.CNTL      Col 1      Format CHAR
                                SYSIPT data NO

1...5...10...5...20...5...30...5...40...5...50...5...60...5...70...
00091 // EXEC LIBR,PARM='MSHP'
00092      DELETE SUBLIB=CNXV8R9.CICS
00093      DELETE SUBLIB=CNXV8R9.CNTL
00094      DELETE SUBLIB=CNXV8R9.COPYBOOK
00095      DELETE SUBLIB=CNXV8R9.OBJLIB
00096      DELETE SUBLIB=CNXV8R9.RUNLIB
00097      DELETE SUBLIB=CNXV8R9.SOURCE
00098 /*
00099 // EXEC IDCAMS,SIZE=AUTO
00100      DELETE CONNX.LIBRARY
00101      CLUSTER PURGE
00102      CATALOG (CNX002.USER.CATALOG)
00103 /*

Command ==>
F1=Help F2=SplitJoin F3=Save+Exit F4=Left F5=Right F6=RFind F7=Bkwd
F8=Fwd F10=RChange F11=CRetrieve F12=Quit
MA b 22/015

```


Step 6 - Install Adabas SQL Gateway

Mainframe Installation of Adabas SQL Gateway

Mainframe Installation

Installation instructions for the Adabas SQL Gateway on an IBM mainframe-compatible server can be found in [Step 5 - Install IBM Mainframe-Compatible Server](#) under either VSE or OS/390, Z/OS, depending on your server type.

Unix Installation

General Information

This step copies the Adabas server components from a client PC to an UNIX target. The CONNX Administrator component must be installed on the client PC.

Note: *All UNIX instructions must be entered as lowercase characters.*

To install the Adabas SQL Gateway server component on a UNIX machine

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the **UNIX/Linux** tab.

The screenshot shows the 'CONNX Server Component Installation' dialog box with the 'UNIX/Linux' tab selected. The dialog has a title bar with standard window controls. Below the title bar is a 'Files' section with the title 'CONNX Server Component Installation' and the instruction 'Enter the platform type, transfer method and login credentials.' To the right of this text is a red puzzle piece icon. Below the instruction are four tabs: 'z/OS', 'VSE', 'VMS', and 'UNIX/Linux' (which is active). The main area is divided into three sections: 'Platform' with a dropdown menu showing 'Linux (Intel) 32-bit'; 'Product' with a grid of checkboxes for ADABAS, CISAM, DISAM, Microfocus, RMCobol, and Dataflex; and 'Login Information' with input fields for Server, Username, Password, and Path (optional). To the right of these is a 'Select transfer method' section with radio buttons for FTP Installation (selected), Secure FTP (sftp) Installation, Secure Copy (scp) Installation, and Manual Copy Installation. At the bottom right are 'Install' and 'Done' buttons.

2. Select your system's platform from the **Platform** drop-down box. Select the **Adabas** check box in the **Product** area.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform

Linux (Intel) 64-bit

Product

☒ ADABAS

☐ DISAM

Login Information

Server

Username

Password

Path (optional)

Select transfer method

☒ FTP Installation

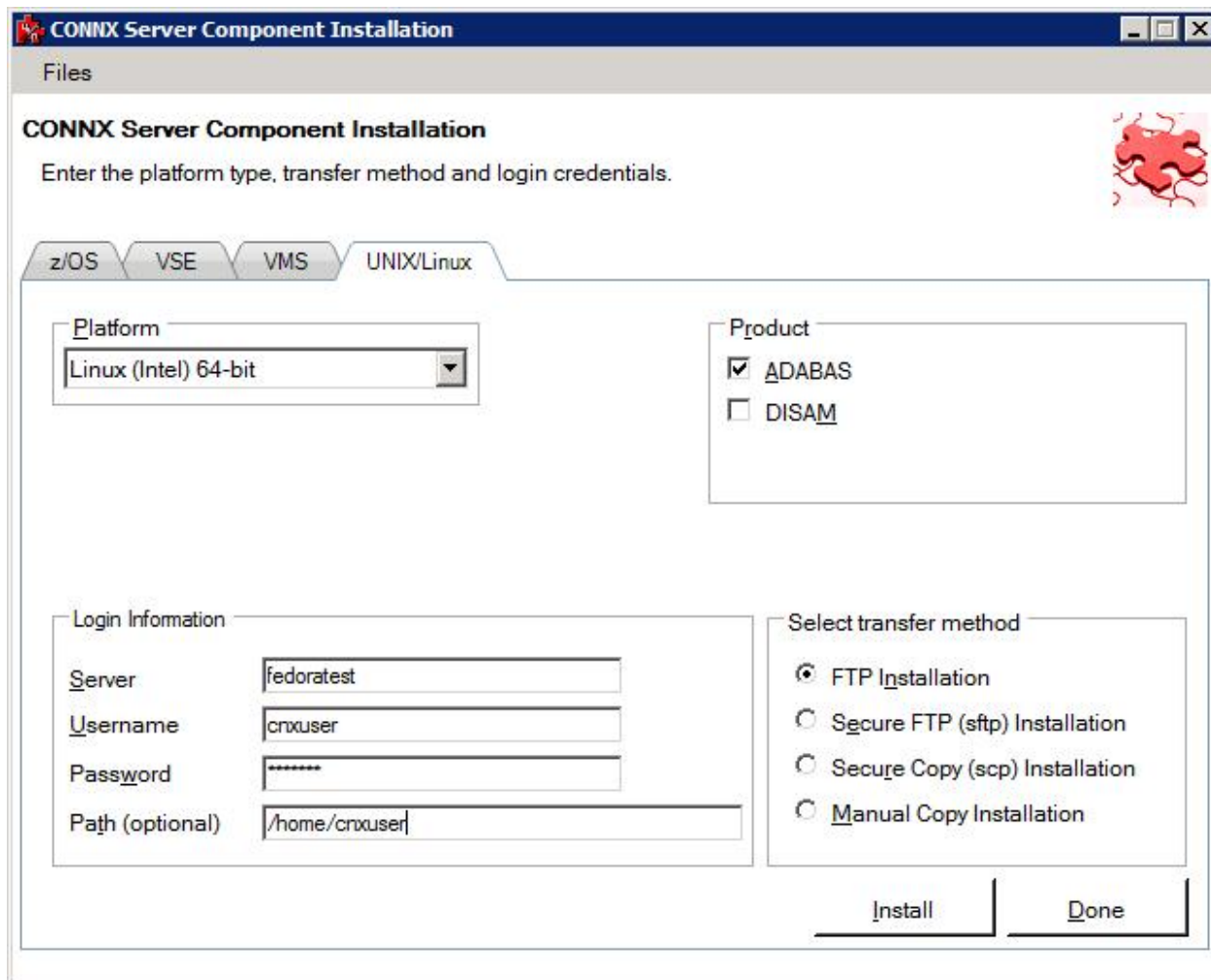
☐ Secure FTP (sftp) Installation

☐ Secure Copy (scp) Installation

☐ Manual Copy Installation

Install Done

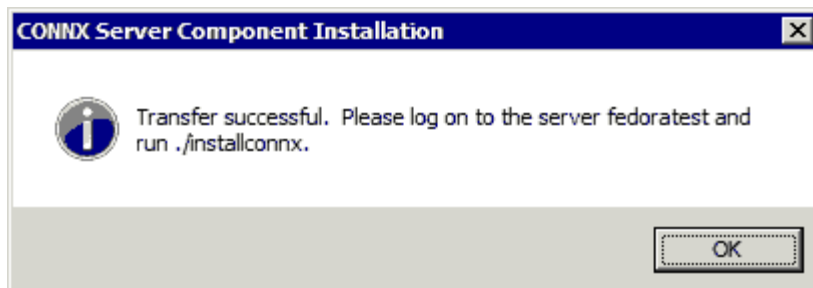
3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the ADABAS\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 12 below.
5. Select the desired **transfer method** and then in the **Server** text box, enter the TCP/IP host name or address for your system platform.
6. In the **User ID** text box, enter a privileged user account name. In the **Password** text box, enter a user account password.
7. By default, the Adabas data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **Installation Information Install Path** text box.



The image shows the 'CONNX Server Component Installation' window. It has a title bar with the text 'CONNX Server Component Installation' and standard window controls. Below the title bar is a 'Files' menu. The main area is titled 'CONNX Server Component Installation' and contains the instruction 'Enter the platform type, transfer method and login credentials.' There are four tabs: 'z/OS', 'VSE', 'VMS', and 'UNIX/Linux'. The 'UNIX/Linux' tab is selected. In the 'Platform' section, a dropdown menu shows 'Linux (Intel) 64-bit'. In the 'Product' section, the 'ADABAS' checkbox is checked, and the 'DISAM' checkbox is unchecked. In the 'Login Information' section, there are four text boxes: 'Server' with 'fedoratest', 'Username' with 'cnxuser', 'Password' with a masked password, and 'Path (optional)' with '/home/cnxuser'. In the 'Select transfer method' section, the 'FTP Installation' radio button is selected, and the other three ('Secure FTP (sftp) Installation', 'Secure Copy (scp) Installation', and 'Manual Copy Installation') are unselected. At the bottom right, there are two buttons: 'Install' and 'Done'.

8. Click the **Install** button. This action starts an FTP, SFT or SCP session and copies the selected components to the target server. Progress will be displayed at the bottom of the screen.

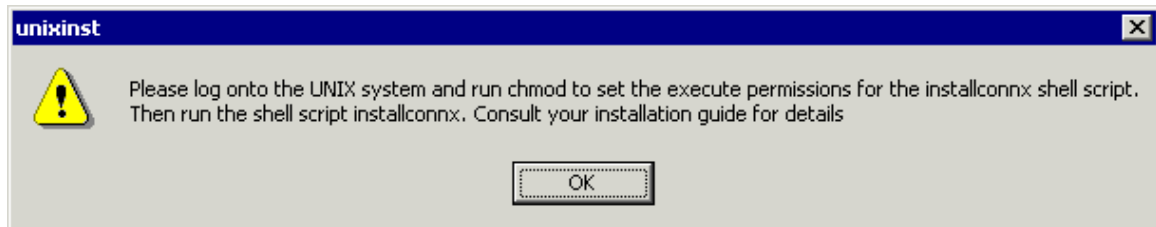
9. An ftp window appears saying the files were successfully copied to the system and instructs you to log on to your system and run the **installconnx** shell script..



10. If Adabas is not installed, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.

Note: If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.

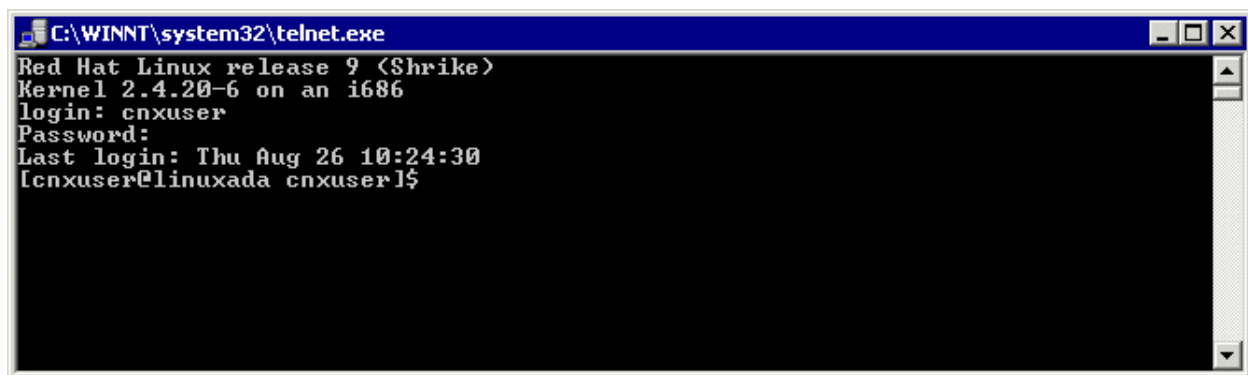
11. Note: If you get this screen, you must use the `chmod` command to set the execute permissions for the `installconnx` shell script.



Example:

`chmod 550 installconnx`

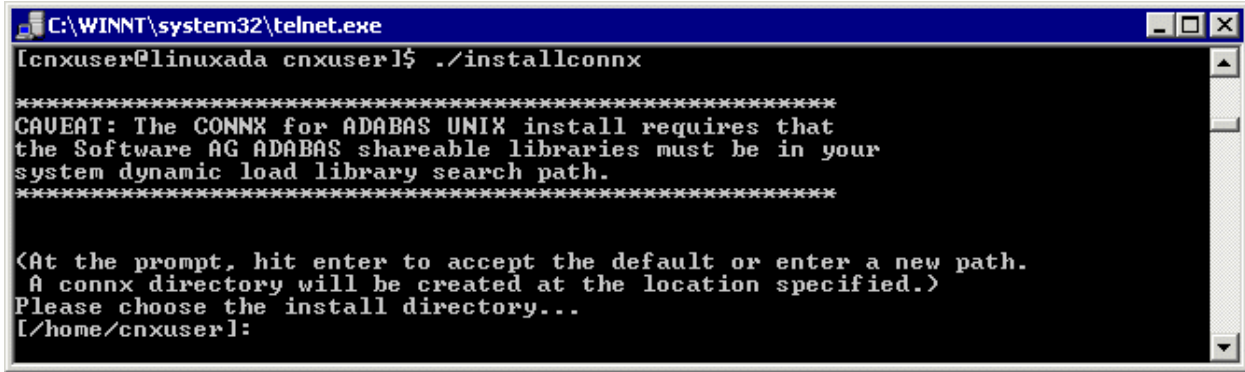
12. To run the **installconnx** shell script, start a terminal emulator on the UNIX server and type the appropriate user name and password.



13. There are three possible command-line options for `installconnx`:

1. `-h` = Print Help.
2. `-s` = Yes | No enhanced security.
3. `-p` = Default CNXRUNPORT.

14. Type **`./installconnx`** and then press **<Enter>**. Press **<Enter>** to install to current directory or enter a new install path at the prompt. The contents of the compressed file are unzipped to a CONNX directory in the install directory.



```

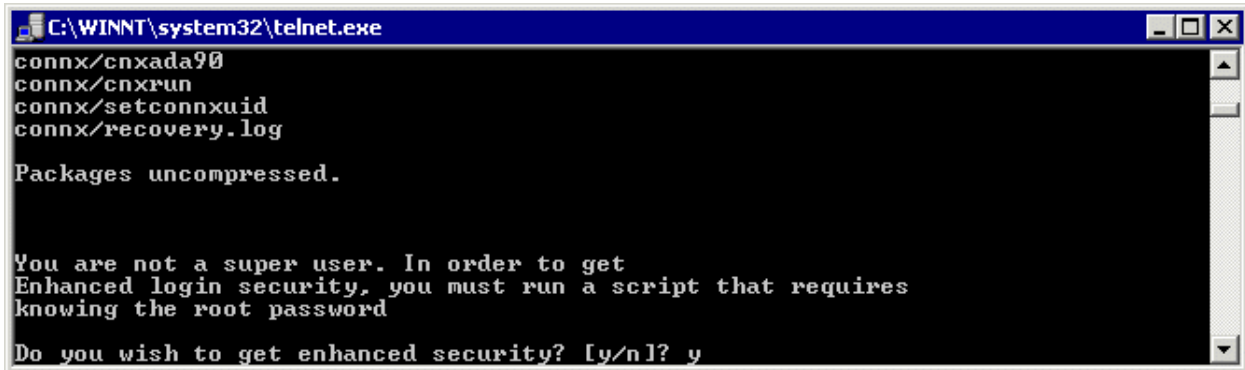
C:\WINNT\system32\telnet.exe
[cnxuser@linuxada cnxuser]$ ./installconnx

*****
CAUEAT: The CONNX for ADABAS UNIX install requires that
the Software AG ADABAS shareable libraries must be in your
system dynamic load library search path.
*****

<At the prompt, hit enter to accept the default or enter a new path.
 A connx directory will be created at the location specified.>
Please choose the install directory...
[/home/cnxuser]:

```

15. When asked whether you wish to get enhanced security, type **<y>**. Enhanced security enables UNIX user names/passwords to be verified using UNIX logon security. You will be prompted for a root password.



```

C:\WINNT\system32\telnet.exe
connx/cnxada90
connx/cnxrun
connx/setconnxuid
connx/recovery.log

Packages uncompressed.

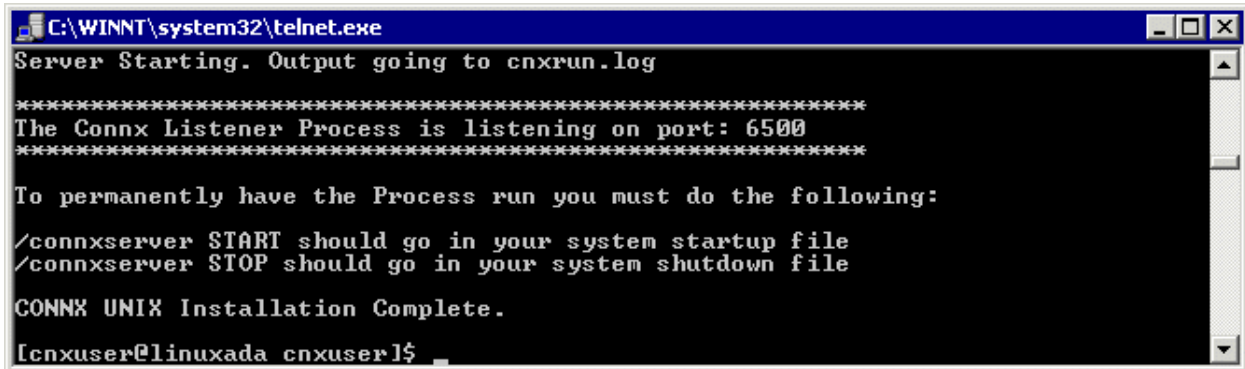
You are not a super user. In order to get
Enhanced login security, you must run a script that requires
knowing the root password

Do you wish to get enhanced security? [y/n]? y

```

16. The final output should read:

CONNX UNIX Installation Complete.



```

C:\WINNT\system32\telnet.exe
Server Starting. Output going to cnxrun.log

*****
The Connx Listener Process is listening on port: 6500
*****

To permanently have the Process run you must do the following:

/connxserver START should go in your system startup file
/connxserver STOP should go in your system shutdown file

CONNX UNIX Installation Complete.

[cnxuser@linuxada cnxuser]$

```

If it reads as follows:

CONNX UNIX Installation Not Complete.

check the preceding error message.

17. Close the telnet server session.
18. Click the **Done** button.

Note: *All UNIX instructions must be entered as lowercase characters.*

Note: If Adabas is upgraded after the initial install of the Adabas SQL Gateway, the Adabas SQL Gateway MUST be re-installed. If the installation files are still on the Unix/Linux system, you can start at step 12.

To build a Docker image for the Adabas SQL Gateway server component on a Linux machine

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the **UNIX/Linux** tab.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform
Linux (Intel) 64-bit ☐ Docker

Product
☐ ADABAS
☐ CISAM
☐ DISAM

Login Information
Server
Username
Password
Path (optional)

Select transfer method
☒ FTP Installation
☐ Secure FTP (sftp) Installation
☐ Secure Copy (scp) Installation
☐ Manual Copy Installation

Install Done

2. Select your system's platform from the **Platform** drop-down box. Select the **Adabas** check box in the **Product** area. Select the **Docker** checkbox.

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform
Linux (Intel) 64-bit ☒ Docker

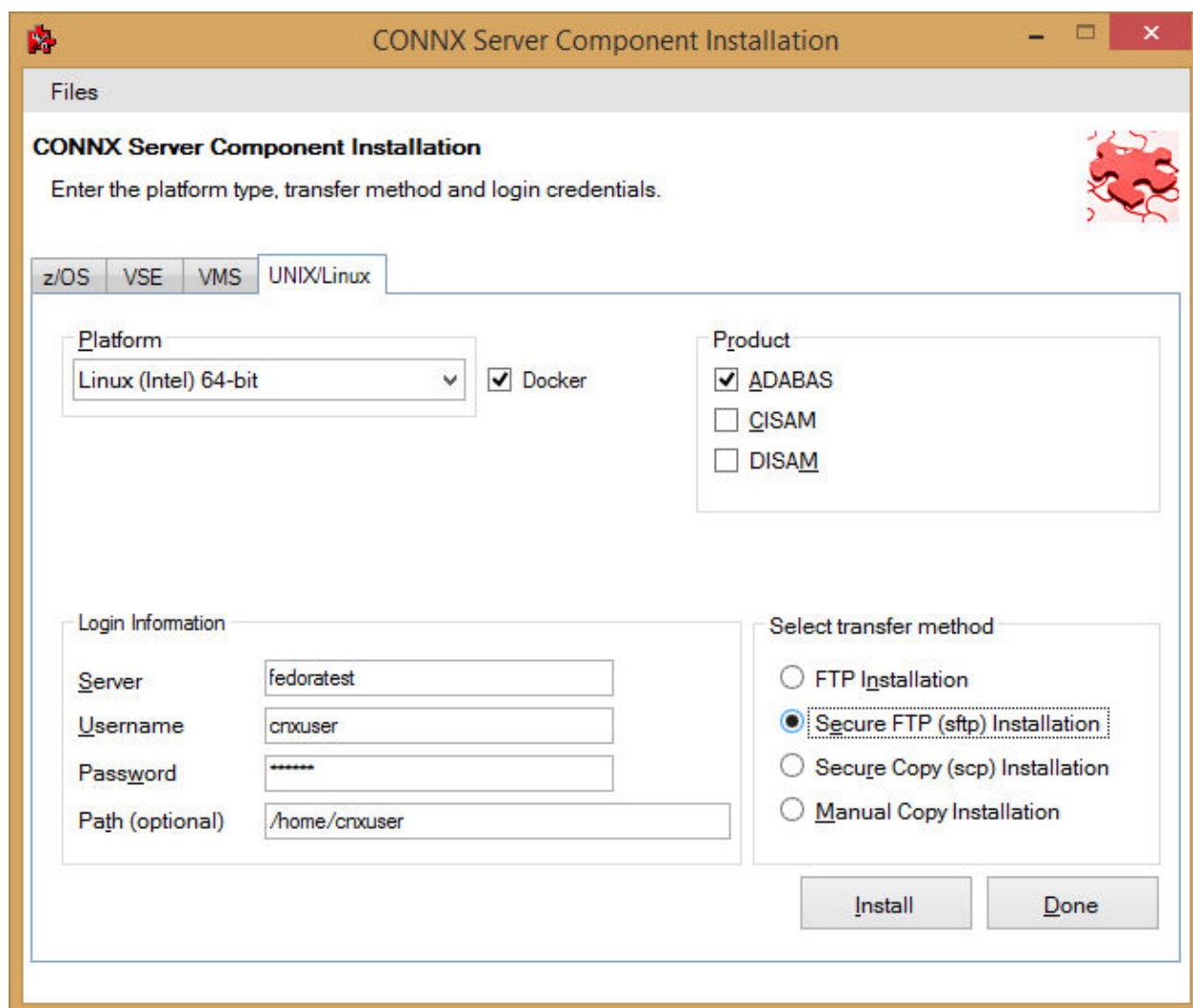
Product
☒ ADABAS
☐ CISAM
☐ DISAM

Login Information
Server
Username
Password
Path (optional)

Select transfer method
☒ FTP Installation
☐ Secure FTP (sftp) Installation
☐ Secure Copy (scp) Installation
☐ Manual Copy Installation

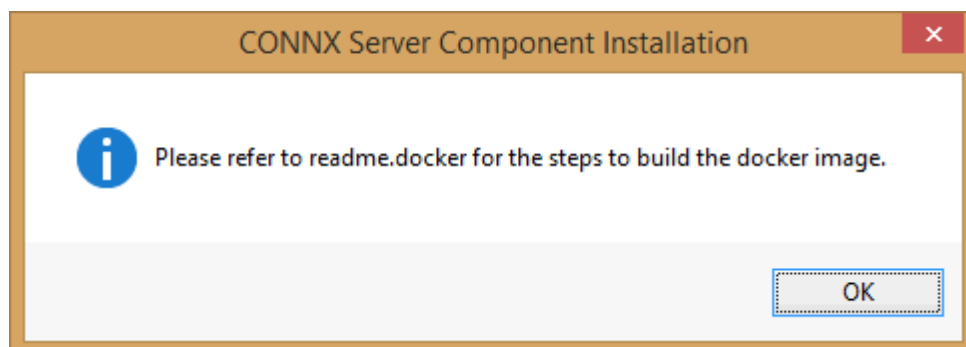
Install Done

3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the ADABAS\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 12 below.
5. Select the desired **transfer method** and then in the **Server** text box, enter the TCP/IP host name or address for your system platform.
6. In the **User ID** text box, enter a privileged user account name. In the **Password** text box, enter a user account password.
7. By default, the Adabas data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **Installation Information Install Path** text box.



The image shows the 'CONNX Server Component Installation' window. It has a title bar with a red icon and standard window controls. Below the title bar is a 'Files' section with the title 'CONNX Server Component Installation' and the instruction 'Enter the platform type, transfer method and login credentials.' To the right of this text is a red puzzle piece icon. Below the instruction are four tabs: 'z/OS', 'VSE', 'VMS', and 'UNIX/Linux'. The 'UNIX/Linux' tab is selected. Inside this tab, there are two main sections. The left section is for 'Platform' and 'Product'. The 'Platform' dropdown is set to 'Linux (Intel) 64-bit'. Next to it is a checked checkbox for 'Docker'. The 'Product' section has three checkboxes: 'ADABAS' (checked), 'CISAM' (unchecked), and 'DISAM' (unchecked). The right section is for 'Login Information' and 'Select transfer method'. The 'Login Information' section has four fields: 'Server' (fedoratest), 'Username' (cnxuser), 'Password' (masked with dots), and 'Path (optional)' (/home/cnxuser). The 'Select transfer method' section has four radio buttons: 'FTP Installation' (unchecked), 'Secure FTP (sftp) Installation' (checked), 'Secure Copy (scp) Installation' (unchecked), and 'Manual Copy Installation' (unchecked). At the bottom right are two buttons: 'Install' and 'Done'.

8. Click the **Install** button. This action starts an FTP, SFTP or SCP session and copies the selected components to the target server. Progress will be displayed at the bottom of the screen.
9. An ftp window appears saying the files were successfully copied to the system and instructs you to log on to your system and follow the instructions in the readme.docker text file..



10. If the file transfer was not successful, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.

Note: If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.

11. To build the Docker image, start a terminal emulator on the Linux server and type the appropriate user name and password.

12. Navigate to the directory you specified in the windows file transfer GUI. This directory will contain several files, including the Docker script file.

13. To build a docker image for the CONNX Adabas Data server, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker build -t connxXXXdataserver . -f./Dockerfile
```

Example:

```
docker build -t connx138dataserver . -f./Dockerfile
```

14. The final output should read:

Successfully tagged connxXXXdataserver:latest.

15. To run the CONNX Data server docker image, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v <adabas client config directory>:/config:z -p <external port>:6500 connxXXXdataserver
```

Example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v /home/cnxuser/adaconfig:/config:z -p 20003:6500 connx138dataserver
```

The <adabas client config directory> should contain your dbmapping.txt to configure TCP/IP connectivity to Adabas nucleus.

Please see Adabas Client documentation for more details.

Ensure that the directory on the host system has an owner and group of sagadmin with an id of 1724. The <external port> should be the port number for connecting to the JDBC server.

16. Verifying the Build:

Please refer to the following docker command reference to manage and verify the image and container.

Show the available docker images:

```
docker images
```

Start the docker image - for example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v /home/cnxuser/adaconfig:/config:z -p
```

20003:6500 connx138dataserver

Show the log:

`docker logs cnxserver`

Show the containers:

`docker ps -a`

Stop the container:

`docker stop cnxserver`

Delete the container:

`docker rm cnxserver`

Remove the image:

`docker image rm connx138dataserver`

UNIX Environment Variables

UNIX configuration settings (also known as environmental variables) are optional and can be added to your connxserver script in the CONNX install directory.

For more information, see CONNX Configuration Settings in the CONNX User Reference Guide, available online, on your CONNX CD-ROM, or within the CONNX product.

To stop and restart the Adabas SQL Gateway server component

Important: This step must be taken if more than one server is running on the same machine to avoid conflicts between servers.

1. To stop a server running on port 6500, enter the following command:
`sh connxserver STOP`
2. To stop a server running on a specific port, enter the following command:
`sh connxserver STOP <port number>`
3. To stop all servers and restart on port 6500, enter the following command:
`sh connxserver START`
4. To stop a server running on one port and restart on another port, enter the following command:
`sh connxserver START <port number>`
5. To stop all child processes that are attached to the server, the CANCEL command must be used. This will eliminate all child processes that are spawned by the server when a new connection is made. The CANCEL command will also stop the main server. To accomplish this, enter the following command:

`sh connxserver CANCEL <port number>`

6. If status is desired, which will show the server and child processes that are running, enter the following command:

`sh connxserver STATUS <port number>`

To edit the connxserver shell script

- Modify the values in the [connxserver](#) script using no spaces between the equal sign (=) and the following value. Add new, defined variables to the export list.

```
#####
# ENVIRONMENT VARS
#
# CNXDIR                = Location of this shell script + the CONNX
#      binaries
# CNXRUNPORT            = Default port (default=6500)
#
#####
CNXDIR=/home/cnxuser/connx
CNXRUNPORT=6500

export CNXDIR CNXRUNPORT
```

Example:

```
#####
# ENVIRONMENT VARS
#
# CNXDIR                = Location of this shell script + the CONNX
#      binaries
# CNXRUNPORT            = Default port (default=6500)
#
#####
CNXDIR=/home/cnxuser/connx
CNXRUNPORT=6500
ENTIRENETWORK=1

export CNXDIR CNXRUNPORT ENTIRENETWORK
```

To edit the Adabas server startup procedure in UNIX

1. On your CONNX administrator computer, start a terminal emulator, and log on to UNIX.
2. Select the desired Adabas UNIX server, and type the appropriate user name and password.
3. Use your favorite UNIX editor (Emacs, for example) to create a shell script from the following text:

```
# begin CONNX startup script
CNXDIR=path to CONNX Adabas server
export CNXDIR
case "$1" in
  start)
    sh $CNXDIR/connxserver START [port number]
    ;;
  stop)
    sh $CNXDIR/connxserver STOP [port number]
    ;;
  restart|reload)
    sh $CNXDIR/connxserver START [port number]
    ;;
esac
exit 0
#end CONNX startup script
```

where [port number] is optional. The default is 6500.

4. Save the document in the appropriate UNIX flavor. The following table contains example file names:

UNIX Flavor	File Name
HP-UX	/sbin/rc2.d/S999connx
LINUX	/etc/rc.d/rc5.d/S99connx
SCO	/etc/rc2.d/S99connx
SUN	/sbin/rc2.d/S99connx
SOLARIS	/sbin/rc2.d/S99connx
TRU-64	/sbin/rc2.d/S99connx
AIX	<p>If you do not have rc#.d directories, where # is the run level number and generally 2 for multi-user, then you can put the following lines toward the bottom of the /etc/rc script. For example:</p> <pre>CNXDIR=path to CONNX Adabas server export CNXDIR sh\$CNXDIR/connxserver START [port number]</pre>

If an example of the file name to use for your UNIX flavor does not appear in the above table, contact CONNX Technical Support for details.

The CONNX Adabas server automatically starts when you restart the system. To verify that it has started, type the following UNIX command after a command prompt in your UNIX system:

```
ps -ef|grep cnxrun
```

5. Check the log file in the CNXDIR directory to verify the start procedure.

Note: *Since the startup procedures vary between operating systems as well as UNIX flavors, the above examples should be used as a rough guide. Consult your operating system's system administration guide for the exact location of startup scripts and directories. You should be logged in as root to complete the script configuration and set the proper permissions on the script.*

Note: *All UNIX instructions must be entered as lowercase characters.*

Installation Tips for the Adabas SQL Gateway (CONNX for Adabas)**Recommended Configurations for the Adabas SQL Gateway (CONNX for Adabas)**

NOTE: These instructions are supplemental to the instructions detailed in the [CONNX Installation Procedure](#).

Installing the CONNX ODBC, OLEDB, .NET Client only

If you are installing the client only, make sure that **Adabas Mainframe Components** is unchecked. Additionally, uncheck the **CONNX Administrator** component for each client. You will need to install the CONNX client components on each computer requiring data access. See [Silent Client Installation](#). It is not necessary to install the administrator component and JDBC server components on machines requiring data access only.

Installing the CONNX JDBC Client only

The CONNX JDBC Client is a pure java JDBC driver, stored in a java jar file. This driver can be obtained by installing the CONNX client components, or by simply copying the CONNXJDBC.JAR file to the desired client, and adding its location to the CLASSPATH. The driver is located in the \connx32\CONNXJDBC\java\jar directory after installing the CONNX client. For more information, see To load and register the CONNX JDBC Driver in the CONNX User Reference Guide.

Installing the CONNX Server components

If you want to install the mainframe version of Adabas, during CONNX installation, you must scroll down to view and select the **Adabas Mainframe Components** check box during installation. The mainframe server components need only be installed once on each mainframe. If you install the CONNX client on four different client machines, for example, you don't need to install the mainframe components again for each client.

For administrator machines (PCs that are authorized to create and modify CONNX data dictionaries), ensure that both the CONNX client driver and the CONNX administrator components are installed. You will need to install the CONNX client components on each computer requiring data access. See [Silent Client Installation](#). It is not necessary to install the administrator component and JDBC server components on machines requiring data access only.

Installing CONNX on a server with Windows version of Adabas

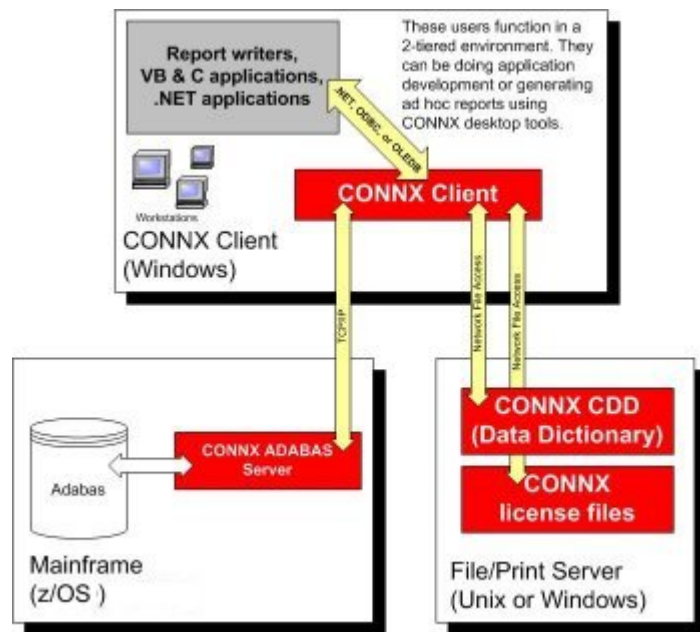
We recommend that you install the CONNX Enterprise Server Service, the CONNX JDBC Server Service, and the CONNX Administrator on the server that contains the Windows Adabas database. This will improve query performance and reduce network traffic between the CONNX clients and the Adabas database.

During installation, make sure that the **Adabas Mainframe Components** check box is unchecked. Check the **CONNX Administrator** component, the **CONNX Enterprise Server Service**, and the **CONNX JDBC Server Service**.

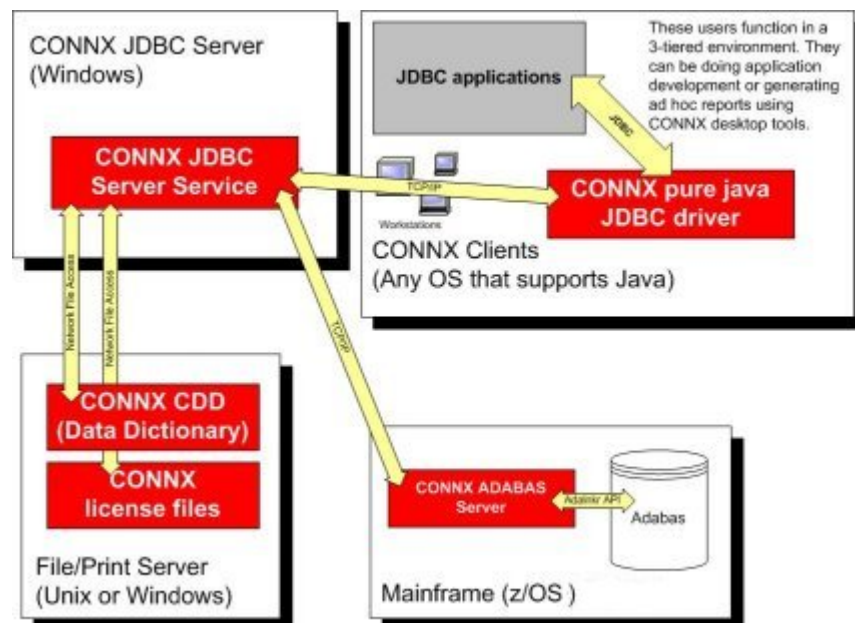
Recommended configurations

Please refer to the following four diagrams for the recommended configurations of the Adabas SQL Gateway.

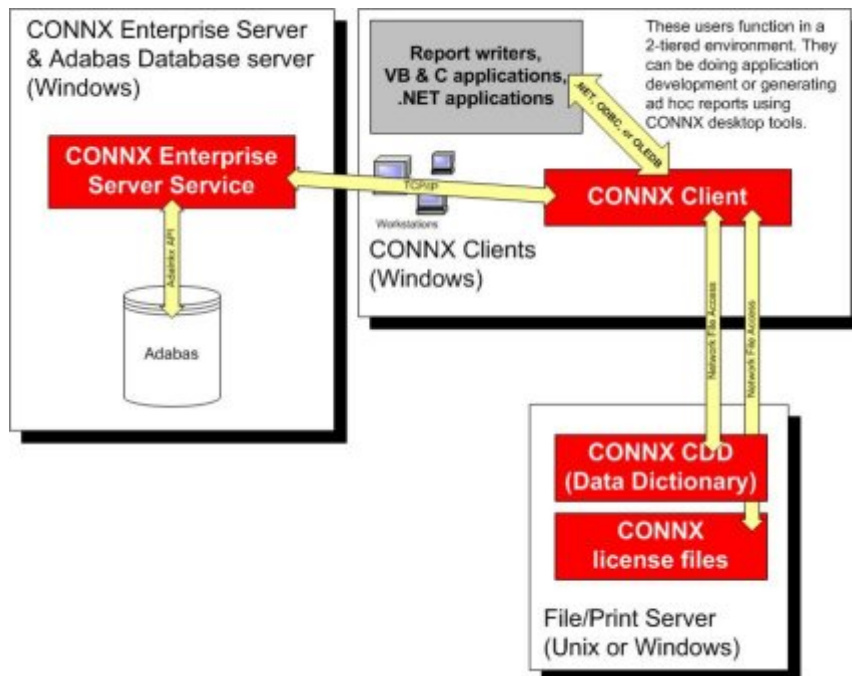
ODBC, OLEDB, and .NET SQL access to Adabas from Windows to the mainframe running z/OS



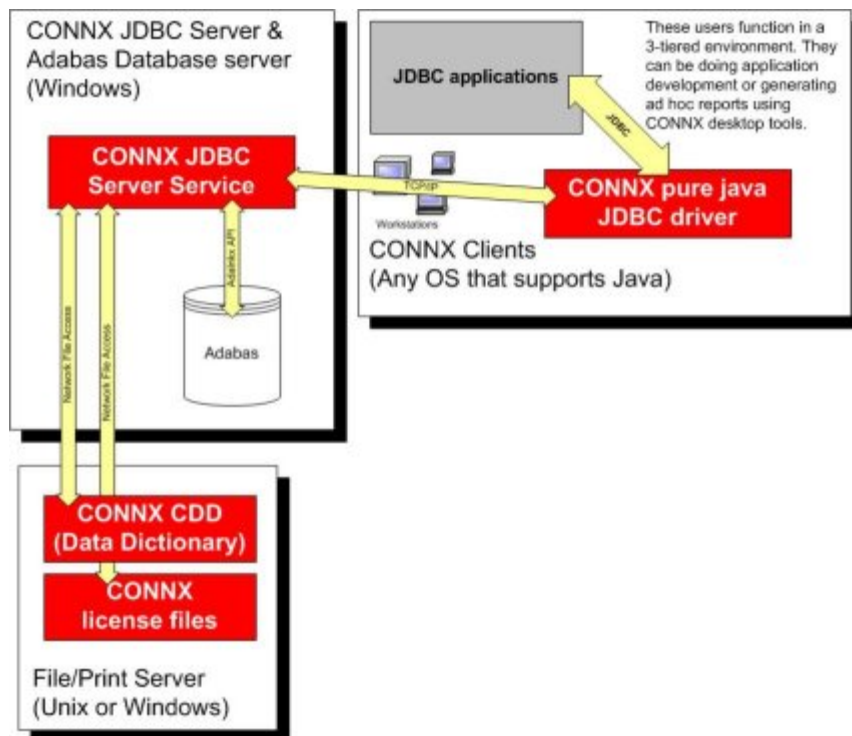
JDBC SQL access to Adabas from anywhere to the mainframe running z/OS



ODBC, OLEDB, and .NET SQL access to Adabas from Windows to a Windows ADABAS database



JDBC SQL access to Adabas from anywhere to a Windows Adabas database



Prerequisites - Mainframe

Adabas SQL Gateway (CONNX for Adabas) requires that:

- The Adabas database product be installed.
- The target system be set up to support TCP/IP communications (OE stack).
- The TCP/IP started task must be running.
- Read/write FTP access be configured and available on the target system for at least one TSO user ID for installation only.
- OSS segment must be defined for the user ID in a security system (RACF / ACF2 / Top Secret) or it must be available by default.

CONNX for Adabas is a reentrant application which executes as a started task or a batch job. During installation, the CONNX Adabas server program must be linked with the reentrant version of the Adabas batch/TSO link routine (ADALNKR). Please see the appropriate Adabas documentation for more information on ADALNKR.

Before submitting the CONNX installation job, please check with your systems programmer or Adabas administrator to ensure that the ADALNKR load module exists in a load library which can be link-edited with the CONNX for Adabas object code.

See [CONNX System Requirements](#) for a more detailed listing.

Prerequisites - Windows

Adabas SQL Gateway, Windows version, requires that:

- You have 40 MB available disk space.
- TCP/IP access.

See [CONNX System Requirements](#) for a more detailed description.

Prerequisites - UNIX

Adabas SQL Gateway, UNIX version, requires that:

- You have 2 MB available disk space.
- TCP/IP access.

See [CONNX System Requirements](#) for a more detailed description.

Step 7 - Install C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus Server (for Unix only)

General Information for C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus

This step copies the CONNX CISAM, DISAM, or Micro Focus server components from a client PC to an UNIX target. The CONNX Administrator component must be installed on the client PC.

If you are not installing the CONNX C-ISAM, DISAM, or Micro Focus module on a Unix target, skip this step and continue to [Step 9 - Post-Installation Checklist](#).

Note: *All UNIX instructions must be entered as lowercase characters.*

To install the CONNX C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus component on a UNIX machine

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the **UNIX/Linux** tab.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform

Linux (Intel) 32-bit

Product

☐ ADABAS ☐ Microfocus

☐ CISAM ☐ RMCobol

☐ DISAM ☐ Dataflex

Login Information

Server

Username

Password

Path (optional)

Select transfer method

☒ FTP Installation

☐ Secure FTP (sftp) Installation

☐ Secure Copy (scp) Installation

☐ Manual Copy Installation

Install Done

2. Select your system's platform from the **Platform** drop-down box. Select the **C-ISAM, DISAM, RMCobol or Micro Focus** check box in the **Product** area.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS UNIX/Linux

Platform

Linux (Intel) 32-bit

Product

☐ ADABAS ☒ Microfocus

☒ CISAM ☒ RMCobol

☒ DISAM ☒ Dataflex

Login Information

Server

Username

Password

Path (optional)

Select transfer method

☒ FTP Installation

☐ Secure FTP (sftp) Installation

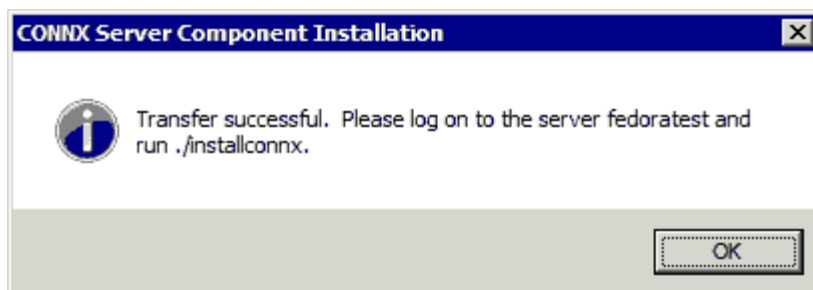
☐ Secure Copy (scp) Installation

☐ Manual Copy Installation

Install Done

3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the <database type>\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 12 below.
5. Select the desired **transfer method** and then in the **Server** text box, enter the TCP/IP host name or address for your system platform.
6. In the **User ID** text box, enter a privileged user account name. In the **Password** text box, enter a user account password.
7. By default, the C-ISAM, DISAM, RMCobol or Micro Focus data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **Installation Information Install Path** text box.

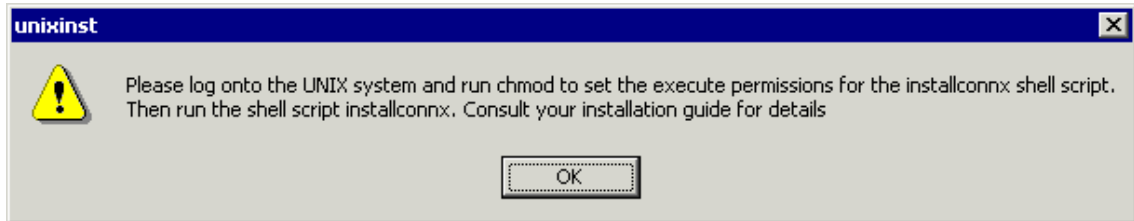
8. Click the **Install** button. This action starts an FTP, SFTP or SCP session and copies the selected components to the target server. Progress will be displayed at the bottom of the screen.
9. An ftp window appears saying the files were successfully copied to the system.



10. If the C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus server is not installed, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.

Note: If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.

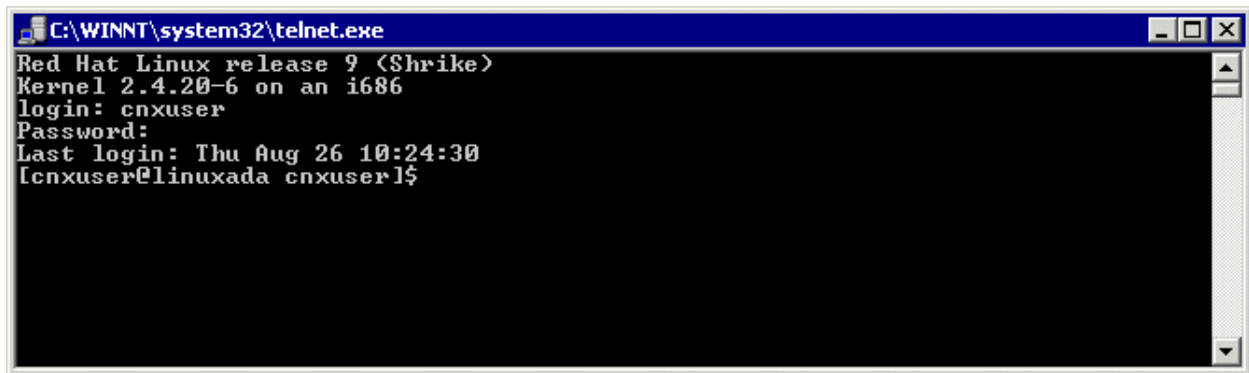
11. Note: If you get this screen, you must use the `chmod` command to set the execute permissions for the `installconnx` shell script.



Example

```
chmod 550 installconnx
```

12. To run the **installconnx** shell script, start a terminal emulator on the UNIX server and type the appropriate user name and password.



13. There are three possible command-line options for `installconnx`:

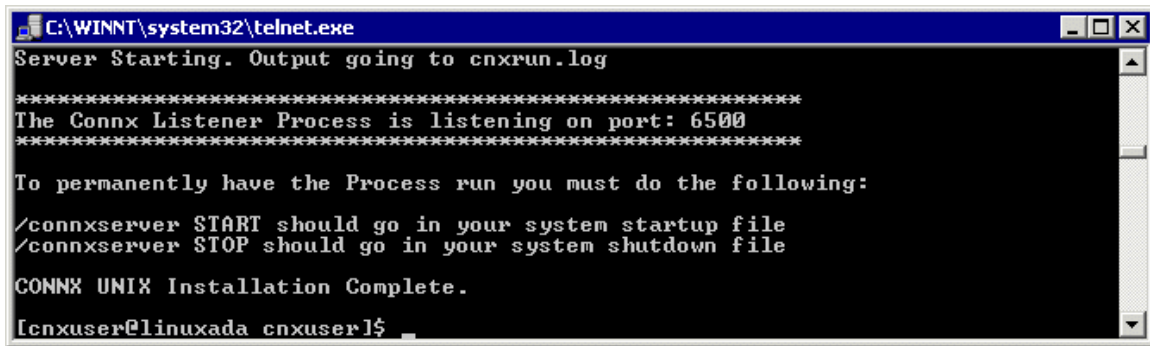
1. `-h` = Print Help.
2. `-s` = Yes | No enhanced security.
3. `-p` = Default CNXRUNPORT.

14. Type `./installconnx` and then press **<Enter>**. Press **<Enter>** to install to current directory or enter a new install path at the prompt. The contents of the compressed files are unzipped to a CONNX directory in the install directory.

15. When asked whether you wish to get enhanced security, type **<y>** Enhanced security enables UNIX user names/passwords to be verified using UNIX logon security. You will be prompted for a root password.

16. The final output should read:

CONNX UNIX Installation Complete.



```

C:\WINNT\system32\telnet.exe
Server Starting. Output going to cnxrun.log

*****
The Connx Listener Process is listening on port: 6500
*****

To permanently have the Process run you must do the following:
/connxserver START should go in your system startup file
/connxserver STOP should go in your system shutdown file

CONNX UNIX Installation Complete.

[cnxuser@linuxada cnxuser]$
  
```

If it reads as follows:

CONNX UNIX Installation Not Complete.

check the preceding error message.

15. Close the telnet server session.
16. If you are installing the DISAM component, ensure that the risamsrv daemon is running.
17. Click the **Done** button.

Note: All UNIX instructions must be entered as lowercase characters.

To build a Docker image for the C-ISAM and D-ISAM server component on a Linux machine

1. Select a computer on which the CONNX Administrator component is installed. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Server Setup**. The CONNX Server Component Installation dialog box appears. Select the **UNIX/Linux** tab.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform

Linux (Intel) 64-bit ☐ Docker

Product

☐ ADABAS
☐ CISAM
☐ DISAM

Login Information

Server
Username
Password
Path (optional)

Select transfer method

☒ FTP Installation
☐ Secure FTP (sftp) Installation
☐ Secure Copy (scp) Installation
☐ Manual Copy Installation

Install Done

2. Select your system's platform from the **Platform** drop-down box. Select the **CISAM** or **DISAM** check box in the **Product** area. Select the **Docker** check box.

CONNX Server Component Installation

Files

CONNX Server Component Installation

Enter the platform type, transfer method and login credentials.

z/OS VSE VMS **UNIX/Linux**

Platform
Linux (Intel) 64-bit ☒ Docker

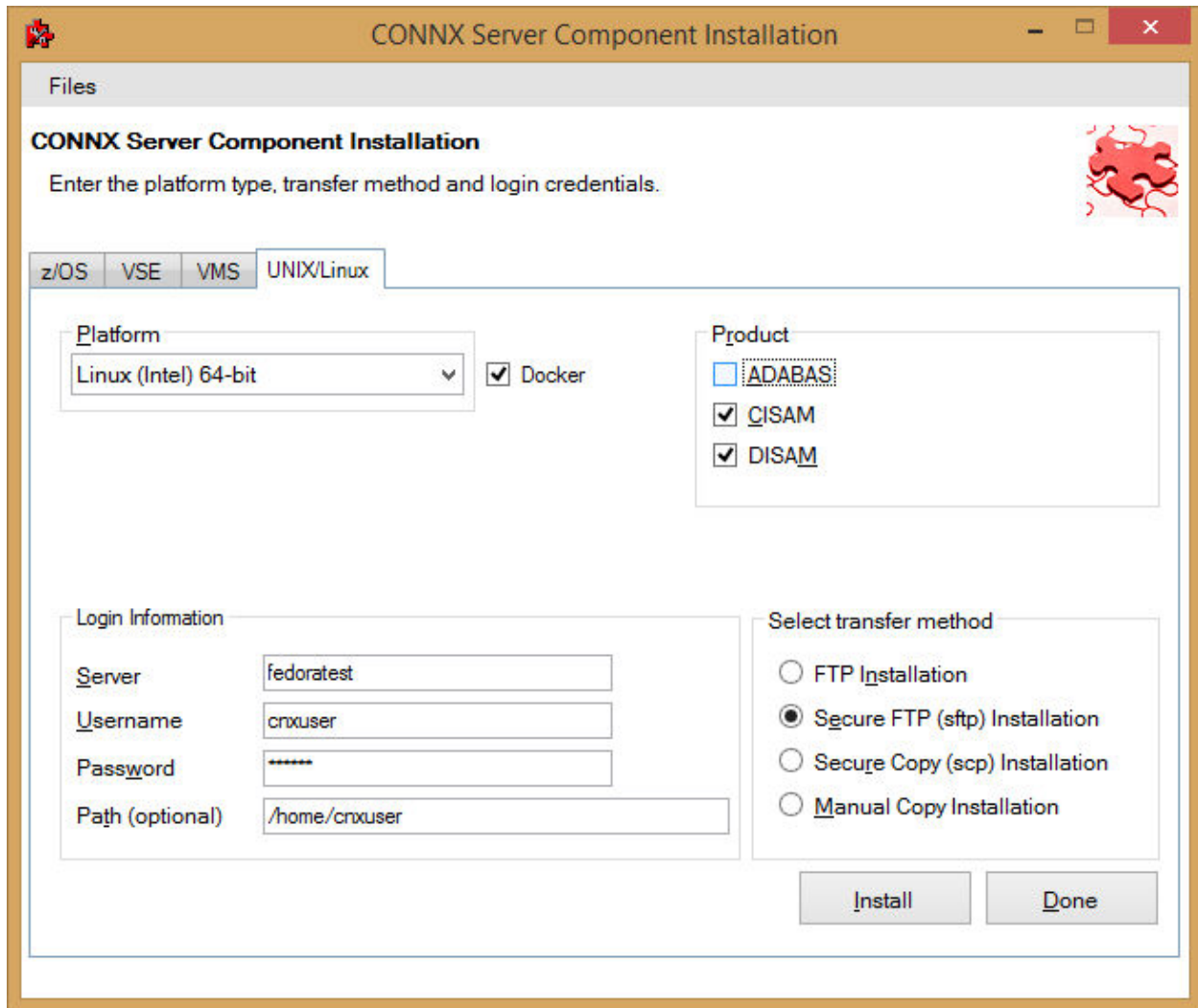
Product
☐ ADABAS
☒ CISAM
☒ DISAM

Login Information
Server
Username
Password
Path (optional)

Select transfer method
☒ **FTP Installation**
☐ Secure FTP (sftp) Installation
☐ Secure Copy (scp) Installation
☐ Manual Copy Installation

3. If the target system has an FTP, SFTP or SCP server enabled, skip to step 5.
4. If the target system does not have an FTP, SFTP or SCP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the CISAM\TEMPINST or DISAM\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to step 12 below.
5. Select the desired **transfer method** and then in the **Server** text box, enter the TCP/IP host name or address for your system platform.
6. In the **User ID** text box, enter a privileged user account name. In the **Password** text box, enter a user account password.
7. By default, the Adabas data is sent to your home directory. If you wish to send it to another location, enter the name of the alternate directory in the **Installation Information Install Path** text box.

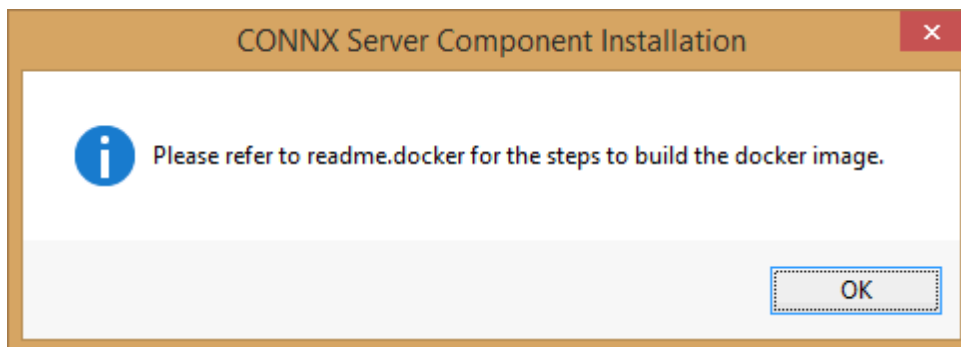
Step 7 - Install C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus Server (for Unix only)



The image shows a window titled "CONNX Server Component Installation". At the top, it says "Files" and "CONNX Server Component Installation". Below that, it says "Enter the platform type, transfer method and login credentials." There are four tabs: "z/OS", "VSE", "VMS", and "UNIX/Linux". The "UNIX/Linux" tab is selected. In the "Platform" section, there is a dropdown menu showing "Linux (Intel) 64-bit" and a checked checkbox for "Docker". In the "Product" section, there are three checkboxes: "ADABAS" (unchecked), "CISAM" (checked), and "DISAM" (checked). In the "Login Information" section, there are four fields: "Server" (fedoratest), "Username" (cnxuser), "Password" (masked with asterisks), and "Path (optional)" (/home/cnxuser). In the "Select transfer method" section, there are four radio buttons: "FTP Installation" (unchecked), "Secure FTP (sftp) Installation" (checked), "Secure Copy (scp) Installation" (unchecked), and "Manual Copy Installation" (unchecked). At the bottom right, there are two buttons: "Install" and "Done".

8. Click the **Install** button. This action starts an FTP, SFTP or SCP session and copies the selected components to the target server. Progress will be displayed at the bottom of the screen.

9. An ftp window appears saying the files were successfully copied to the system and instructs you to log on to your system and follow the instructions in the readme.docker text file..



10. If the file transfer was not successful, an error box appears. Select **Files** from the menu. The sub menu has options to view the FTP log as well as the FTP script. The FTP logs have a timestamp appended to the name. Select the appropriate log file and check for errors. If the resolution to the problem requires a call to technical support, please provide the log file as well as the script file to the support representative.

Note: If the files fail to copy via the Client Installation's FTP, SFTP or SCP session, please use the Manual Copy Installation as outlined in step 4 above.

11. To build the Docker image, start a terminal emulator on the Linux server and type the appropriate user name and password.

12. Navigate to the directory you specified in the windows file transfer GUI. This directory will contain several files, including the Docker script file.

13. To build a docker image for the CONNX Adabas Data server, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker build -t connxXXXdataserver . -f./Dockerfile
```

Example:

```
docker build -t connx138dataserver . -f./Dockerfile
```

14. The final output should read:

Successfully tagged connxXXXdataserver:latest.

15. To run the CONNX Data server docker image, use the following command:

(replace XXX with the current CONNX version number - 138 for example for 13.8)

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v <data directory>:/data:z -p <external port>:6500 connxXXXdataserver
```

Example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v /home/cnxuser/data:/data:z -p 20003:6500 connx138dataserver
```

The <data directory> should contain your C-ISAM or D-ISAM data files (If you are accessing C-ISAM or D-ISAM).

Ensure that the directory, and all data files in that directory on the host system have an owner and group of sagadmin with an id of 1724.

The <external port> should be the port number for connecting to the JDBC server.

16. Verifying the Build:

Please refer to the following docker command reference to manage and verify the image and container.

Show the available docker images:

```
docker images
```

Start the docker image - for example:

```
docker run -d -e "ACCEPT_EULA=Y" --name cnxserver -v /home/cnxuser/data:/data:z -p 20003:6500
```

Step 7 - Install C-ISAM, DISAM, RMCobol, Dataflex and Micro Focus Server (for Unix only)

connx138dataserver

Show the log:

```
docker logs cnxserver
```

Show the containers:

```
docker ps -a
```

Stop the container:

```
docker stop cnxserver
```

Delete the container:

```
docker rm cnxserver
```

Remove the image:

```
docker image rm connx138dataserver
```

To edit the C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus server startup procedure

1. On your CONNX administrator computer, start a terminal emulator, and log on to UNIX.
2. Select the desired C-ISAM, DISAM, or Micro Focus server, and type the appropriate user name and password.
3. Use your favorite UNIX editor (Emacs, for example) to create a shell script from the following text:

```
# begin CONNX startup script
CNXDIR=path to CONNX <C-ISAM, DISAM, or Micro Focus> server
export CNXDIR
case "$1" in
  start)
    sh $CNXDIR/connxserver START [port number]
    ;;
  stop)
    sh $CNXDIR/connxserver STOP [port number]
    ;;
  restart|reload)
    sh $CNXDIR/connxserver START [port number]
    ;;
esac
exit 0
#end CONNX startup script
```

where [port number] is optional. The default is 6500.

4. Save the document in the appropriate UNIX flavor. The following table contains example file names:

UNIX Flavor	File Name
HP-UX	/sbin/rc2.d/S999connx
LINUX	/etc/rc.d/rc5.d/S99connx
SCO	/etc/rc2.d/S99connx
SUN	/sbin/rc2.d/S99connx
SOLARIS	/sbin/rc2.d/S99connx
TRU-64	/sbin/rc2.d/S99connx
AIX	<p>If you do not have rc#.d directories, where # is the run level number and generally 2 for multi-user, then you can put the following lines toward the bottom of the /etc/rc script. For example:</p> <pre>CNXDIR=path to CONNX <C-ISAM, DISAM, or Micro Focus> server export CNXDIR sh\$CNXDIR/connxserver START [port number]</pre>

--	--

If an example of the file name to use for your UNIX flavor does not appear in the above table, contact CONNX Technical Support for details.

The CONNX C-ISAM, DISAM, or Micro Focus server automatically starts when you restart the system. To verify that it has started, type the following UNIX command after a command prompt in your UNIX system:

```
ps -ef|grep cnxrun
```

5. Check the log file in the CNXDIR directory to verify the start procedure.

Note: *Since the startup procedures vary between operating systems as well as UNIX flavors, the above examples should be used as a rough guide. Consult your operating system's system administration guide for the exact location of startup scripts and directories. You should be logged in as root to complete the script configuration and set the proper permissions on the script.*

Note: *All UNIX instructions must be entered as lowercase characters.*

To stop and restart the C-ISAM, DISAM, RMCobol, Dataflex or Micro Focus server

Important: *This step must be taken if more than one server is running on the same machine to avoid conflicts between servers.*

1. Stop a server by entering the following command:
`sh connxserver STOP`
2. To stop a server running on a specific port, enter the following command:
`sh connxserver STOP <port number>`
3. To stop all servers and restart on port 6500, enter the following command:
`sh connxserver START`
4. To stop a server running on one port and restart on another port, enter the following command:
`sh connxserver START <port number>`
5. Press **<Enter>**.

Step 8 - DB2 Host Configuration

CONNX and DB2

CONNX DB2 Dynamic SQL Packages

The Build Packages text box in the CDD import utility instructs the CONNX DB2 ODBC driver to build a user-specified number **N** of package sets for all isolation levels for two types of cursor behavior: Delete/Close (non-sticky) and Open ("hold" or "sticky"). This translates into the following:

$N * 5 * 2 = 10$ dynamic SQL packages (default for N is 1; maximum is 32)

where

N is the number of identical packages within each set

5 is the number of isolation levels (OS/400 only; 4 for DB2, MVS, and UDB)

2 is the number of types of cursor behavior

320 is the maximum number of dynamic SQL packages built by CONNX DB2

Each package consists of 32 sections. The first package for isolation level No Commit with cursor behavior = CLOSE (CONXNC00) defines the following skeleton cursors:

```
DECLARE NC0001 CURSOR FOR S01
```

```
DECLARE NC0002 CURSOR FOR S02
```

```
DECLARE NC0003 CURSOR FOR S03
```

```
...
```

```
DECLARE NC0032 CURSOR FOR S32
```

Each skeleton cursor is used by the CONNX DB2 ODBC driver to maintain a separate dynamic SQL context, for example:

Select col1 from table maps to cursor 1

Select col1, col2 from table maps to cursor 2

Insert into table (col1,col2) values ('row01', 1) maps to cursor 3

Delete from table where col1 = 'row01' maps to cursor 4

For each isolation level and cursor behavior, 1-32 identical SQL packages can be defined for use by the CONNX DB2 ODBC driver. This provides a theoretical maximum number of unique dynamic SQL contexts = $32 * 32 = 1024$ **for each isolation level within a unit of work** for cursor behavior = CLOSE, or **for the life of each ODBC connection** for cursor behavior = OPEN.

For example, an ODBC application could connect to the CONNX DB2 ODBC driver, set AutoCommit mode off, and process any mix of 1024 unique DML (SELECT, INSERT, UPDATE, DELETE) or DDL (CREATE, DROP) SQL statements within a unit of work or during the life of a connection.

For data source (CDD) cursor behavior = CLOSE, when the ODBC application terminates the unit of work via an ODBC 2.x or 3.x SQLTransact/SQLEndTran API function call, the CONNX DB2 ODBC driver issues a COMMIT or ROLLBACK and returns the package section contexts (section numbers and cursor names) to a pool for reuse.

For data source (CDD) cursor behavior = OPEN or PRESERVE ("hold" or "sticky" cursors), when the ODBC application calls SQLTransact/SQLEndTran, the CONNX DB2 ODBC driver issues a

COMMIT/ROLLBACK and returns all non-cursor based contexts (INSERT/DELETE/UPDATE, etc.) to the free pool.

For ODBC AutoCommit mode (the default), the CONNX DB2 ODBC driver returns the skeleton cursors to the free pool using a least-frequently used (LFU) algorithm.

In practice, the CONNX DB2 ODBC driver uses only the first package from a package set to process dynamic SQL requests issued by an ODBC application. The choice of current dynamic SQL package set is determined by isolation level and commit mode (AutoCommit or otherwise), as set by the ODBC application (for example, Microsoft Access or Visual Basic) via the ODBC SQLSetConnectOption or SQLSetConnectAttr API functions, and data source cursor behavior (defined in the CONNX DB2 CDD). The extra packages for each package set are defined to support long-running ODBC applications which reuse the same connection, for example, a web server application using an anonymous login such as IUSR_CONNX.

Notes:

1. The dynamic SQL packages and skeleton cursors are managed at the host on a per-connection basis. For instance, any number of CONNX DB2 ODBC client applications can prepare different SQL statements into section 1 of package CONXNC00. The separate SQL contexts for section 1 are managed by the IBM DRDA host program (the Application Server or AS).
2. The user specified number of packages per package set is set to 1 by default. You can adjust the number of packages per package set by adding or changing the following entry in the CONNX.INI file located in your Windows directory.

[DATABASES]

DB2MaxPkgSets=n

Where $1 \leq n \leq 32$

3. After changing the DB2MaxPkgSets entry, from the same client machine, you can use the CONNX CDD Import application to build **n** CONNX DB2 Dynamic SQL Packages per package set by checking the 'Build Packages' check box.
4. It is not necessary to redistribute the CONNX CDD file used during the import/build process after building additional CONNX DB2 dynamic SQL packages. The CONNX DB2 driver has auto-detect functionality which determines the current number of packages per package set at connect time.

Important: All CONNX DB2 dynamic SQL package names begin with 'CONX'; hence, the first package built with isolation level = READ_COMMITTED, Access Mode = READ_WRITE, and cursor behavior = del/close is named CONXCS00. The tenth package in this package set is CONXCS09; the thirty-second package in this set is CONXCS0V.

The dynamic SQL packages created by clicking the 'Build Packages' check box are named as follows:

CONNX DB2 Sample Packages

DB2 Platform	Transaction Isolation	Access Mode	Auto Commit	Cursor Behavior	Package Set Suffix
OS/400	READ_UNCOMMITTED	READ_ONLY	ON	DEL/CLOSE	NC0z
				PRESERVE	NC1z
			OFF	DEL/CLOSE	CH0z
				PRESERVE	CH1z

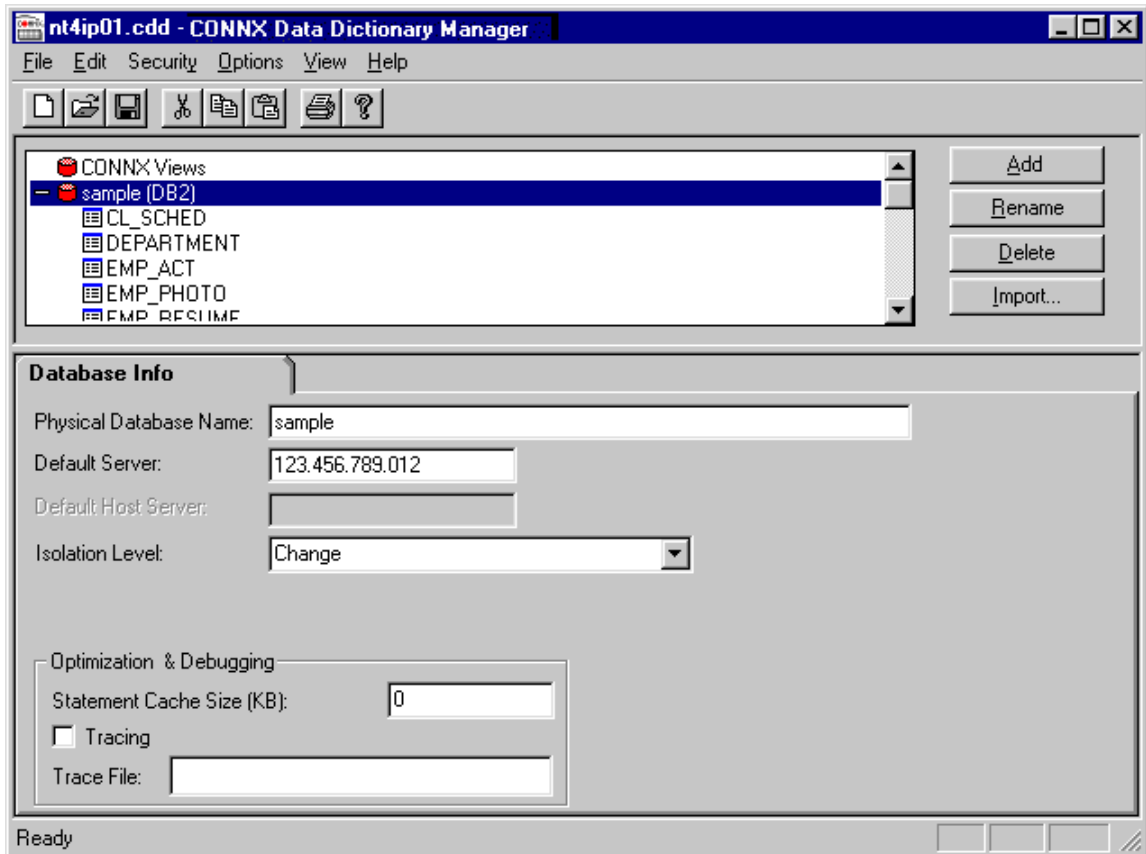
		READ_WRITE	ON	DEL/CLOSE	NC0z
				PRESERVE	NC1z
			OFF	DEL/CLOSE	CH0z
				PRESERVE	CH1z
Non-OS/400	READ_UNCOMMITTED	READ_ONLY	ON	DEL/CLOSE	CH0z
				PRESERVE	CH1z
			OFF	DEL/CLOSE	CH0z
				PRESERVE	CH1z
		READ_WRITE	ON	DEL/CLOSE	CH0z
				PRESERVE	CH1z
			OFF	DEL/CLOSE	CH0z
				PRESERVE	CH1z
All	READ_COMMITTED	READ_ONLY	ON	DEL/CLOSE	CS0z
				PRESERVE	CS1z
			OFF	DEL/CLOSE	CS0z
				PRESERVE	CS1z
		READ_WRITE	ON	DEL/CLOSE	CS0z
				PRESERVE	CS1z
			OFF	DEL/CLOSE	CS0z
				PRESERVE	CS1z
All	REPEATABLE READ	READ_ONLY	ON	DEL/CLOSE	AL0z
				PRESERVE	AL1z
			OFF	DEL/CLOSE	AL0z
				PRESERVE	AL1z
		READ_WRITE	ON	DEL/CLOSE	AL0z
				PRESERVE	AL1z
			OFF	DEL/CLOSE	AL0z
				PRESERVE	AL1z
All	SERIALIZABLE	READ_ONLY	ON	DEL/CLOSE	RR0z

				PRESERVE	RR1z
			OFF	DEL/CLOSE	RR0z
				PRESERVE	RR1z
		READ_WRITE	ON	DEL/CLOSE	RR0z
				PRESERVE	RR1z
			OFF	DEL/CLOSE	RR0z
				PRESERVE	RR1z
Where 0 <= z <= 9 or A <=x<=V					

To establish CONNX and DB2 CDD configuration options

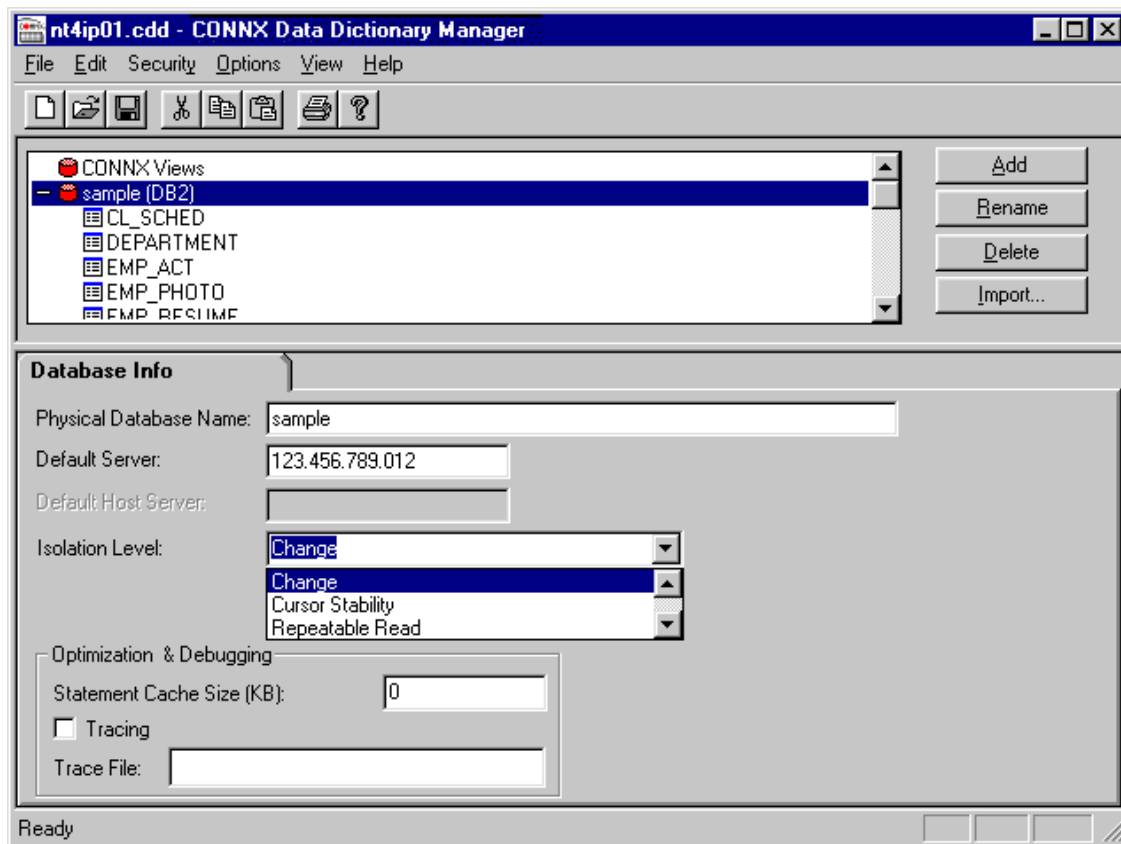
The following procedures may be helpful for troubleshooting connection and configuration options in CONNX.

1. Click on any previously imported DB2 database, e.g. **SAMPLE**, in the CONNX Data Dictionary Manager window.



2. The name of the selected database appears in the **Physical Database Name** text box on the **Database Tab** in the lower pane. The TCP/IP address or SNA location appears in the **Default Server** text box.
3. Select an isolation level in the **Isolation Level** text box.

Selecting the correct isolation level is very important for optimal performance. Note that as you move down the list, concurrency decreases and exclusivity increases. For instance, if the CDD is to be used by a read-only report writer, the isolation level Change (for mainframe or DB2 UDB targets) or No Commit (for the AS/400) is the appropriate choice, since it maximizes the concurrent usage of DB2 objects. Also note that, in most cases, an online transaction processor requires an All or Repeatable Read isolation level, which locks out updates from other clients until the unit of work is complete.



4. Under **Optimization and Debugging**, enter the **Statement Cache Size**, which specifies the maximum number of kilobytes CONNX can use to store cached information related to the most-frequently executed dynamic SQL statements. CONNX keeps track of the most frequently executed dynamic SQL statements in a virtual memory cache. This cache allows CONNX to optimize the length of messages sent to and received from the target host. Depending on the cursor behavior specified in the CONNX CDD, the cache is refreshed either after each COMMIT/ROLLBACK or whenever the cache exceeds its limit. A value of 0 instructs CONNX to allocate a default cache size of 64 KB. For long-running applications such as Web servers, this value can be revised upward to improve performance. To minimize the cache memory footprint, set the value to 32 (not 0).
5. Select the **Tracing** check box to diagnose potential communications problems. If the box is checked and no **Trace File** name is specified, CONNX writes a binary trace of all sent/received messages via either TCP/IP or APPC/SNA LU 6.2 to the **cnxdb2.trc** file in the CONNX32 directory. Simultaneously, CONNX writes a text file (cnxdb2.sql) of all executed SQL statements.
6. Type a file name in the **Trace File** text box to direct the binary trace to a specified file. The SQL text trace is also written to the same directory/file name, with a file extension of .sql.

Turn the feature off after successfully tracing problem scenarios and submitting the binary trace file and/or ODBC API trace to CONNX Technical Support.

CONNX DB2 Module Connection Scenarios

Host Configuration for DB2 Admins

The CONNX DB2 module implements the Distributed Relational Database Architecture (DRDA) application requester functionality, which allows users to connect to any DRDA-based server via either TCP/IP and the Winsock API or by SNA APPC / LU 6.2 (Advanced Program-to-Program Communications via Logical Unit 6.2).

CONNX provides a generic layer that supports the APPC API, which is implemented by many SNA vendors, including Attachmate, IBM, Microsoft, Novell, and others. CONNX can detect these SNA vendor stacks on the installed client machine automatically. If you use an SNA vendor product to communicate to your target DB2 server, then you can set up CONNX to communicate with the DB2 server via APPC LU 6.2 and your SNA client software.

The following procedures describe how to connect to DB2 UDB Windows/OS2/Linux servers, DB2/400, and DB2 OS/390/MVS.

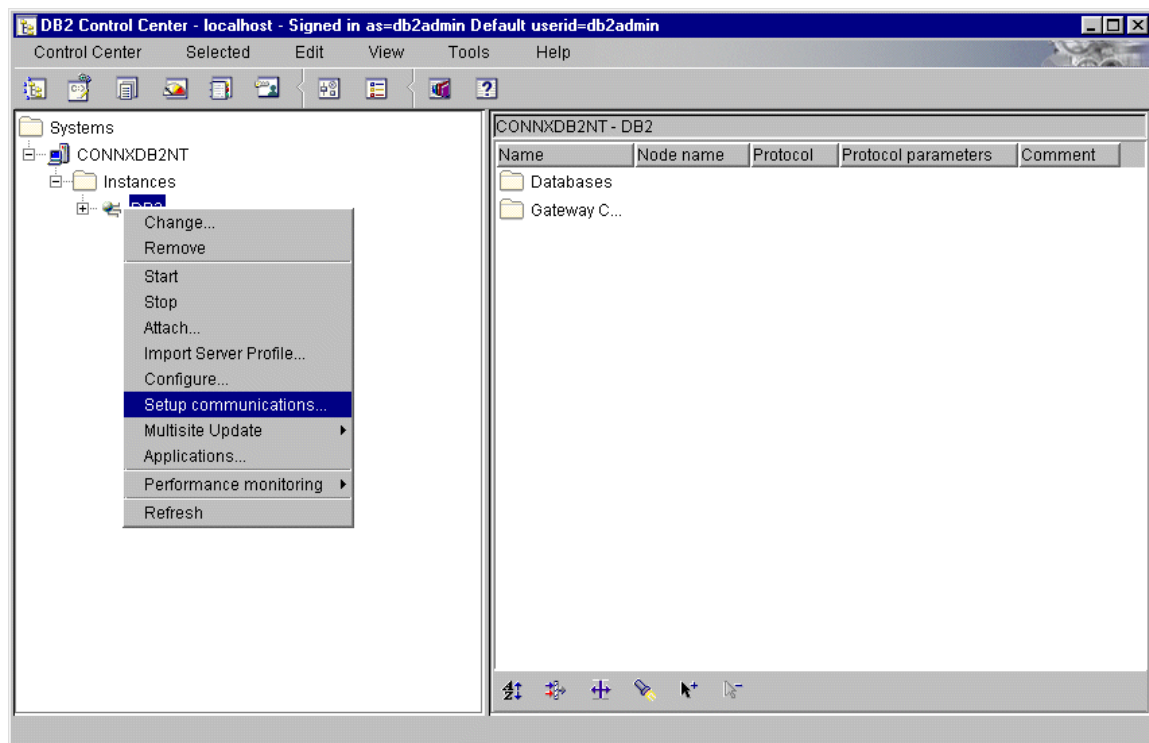
CONNX and DB2/UDB (Universal Database) - TCP/IP

To establish TCP/IP configuration on DB2 UDB

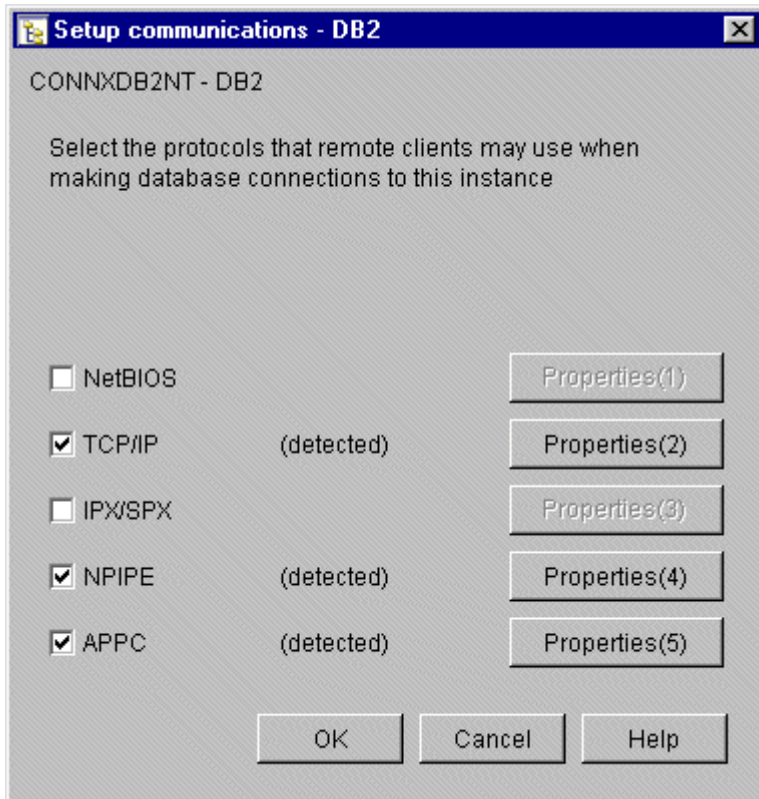
1. After confirming the RDBName, determine the host TCP/IP address and port number.

When DB2 UDB 5.x or 6.x is installed, it auto-detects the TCP/IP stack and updates its configuration and the TCP/IP services file. You can verify the TCP/IP settings via the DB2 Control Center or the DB2 Command Line Processor.

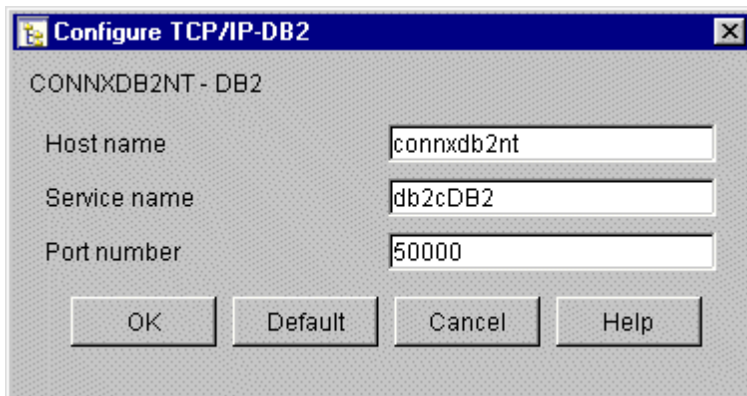
2. Start the **DB2 Control Center**.



3. Click on a DB2 instance in the left pane. Right-click, and select **Setup Communications**.
4. The **Setup Communications – DB2** dialog box appears.



5. If TCP/IP is preconfigured, the **TCP/IP** check box is checked. Click the **Properties** button to the right of the TCP/IP check box in the **Setup Communications** dialog box to display the symbolic host name, service name, and port number.



6. To resolve the symbolic host name into a dotted TCP/IP address, open the hosts file on your local machine, or open a command prompt window and enter the following command:
ping hostname
7. A successful ping returns one or more lines as follows:
Pinging the hostname [123.456.789.012] with 32 bytes of data.

Reply from 123.456.789.012: bytes=32 time<10ms TTL=64
8. You can connect to the target host using either the symbolic host name or the dotted TCP/IP address. This value is entered in the **TCP/IP address** text box in the **CONN CDD Import** dialog box.

To verify server TCP/IP settings via DB2 CLP

1. At the target server, start the DB2 Command Processor window and enter the following command:
Get DBM CFG
2. Verify that the following line appears:
TCP/IP Service Name(SVCENAME) db2cDB2
3. Verify the port on which DB2 listens by opening the services text file in the platform-specific directories,

Windows	\winnt\system32\drivers\etc
AIX	bin/ls/etc

and search for the following entries:

db2cDB2 50000/tcp #connection port for the DB2 instance DB2

db2iDB2 50001/tcp #interrupt port for the DB2 instance DB2

4. If these entries are not present, add them to the services file.
5. Ping the dotted IP address from a Windows client machine by opening a DOS command prompt window and typing the following command:

Ping 123.456.789.012

Where

123.456.789.012 is the TCP/IP address of the DB2 UDB server.

6. If the ping is successful, try to connect from the CONNX **CDD Import** dialog box by entering either the symbolic host name or the dotted IP address of the target DB2 UDB server plus the port number (= 50000, for this example).

To verify server TCP/IP settings for Linux

- Ask your system administrator for the symbolic or 4-part dotted numeric TCP/IP address.

CONNX and DB2/UDB - APPC SNA LU 6.2

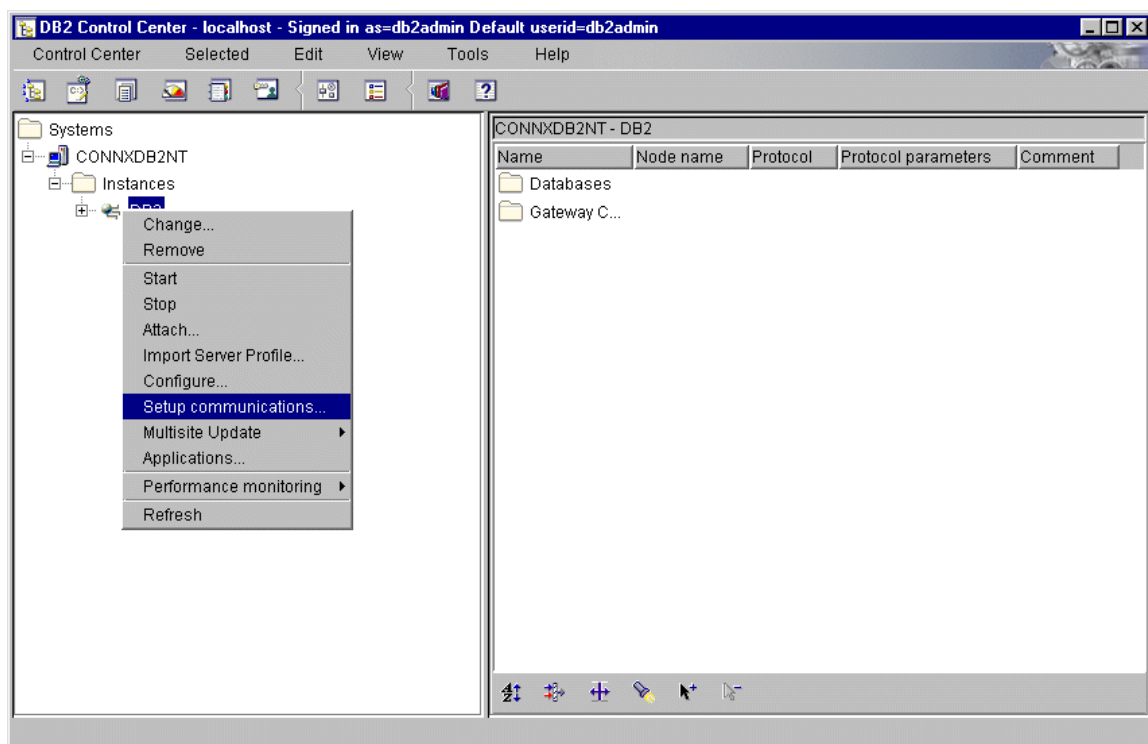
To verify server APPC/LU 6.2 settings via Control Center

The DB2 UDB 5.x and 6.x products can detect selected SNA vendor stacks, including IBM eNetwork Communications Server, automatically. For other SNA vendors, the configurations can be updated using the vendor configuration utility.

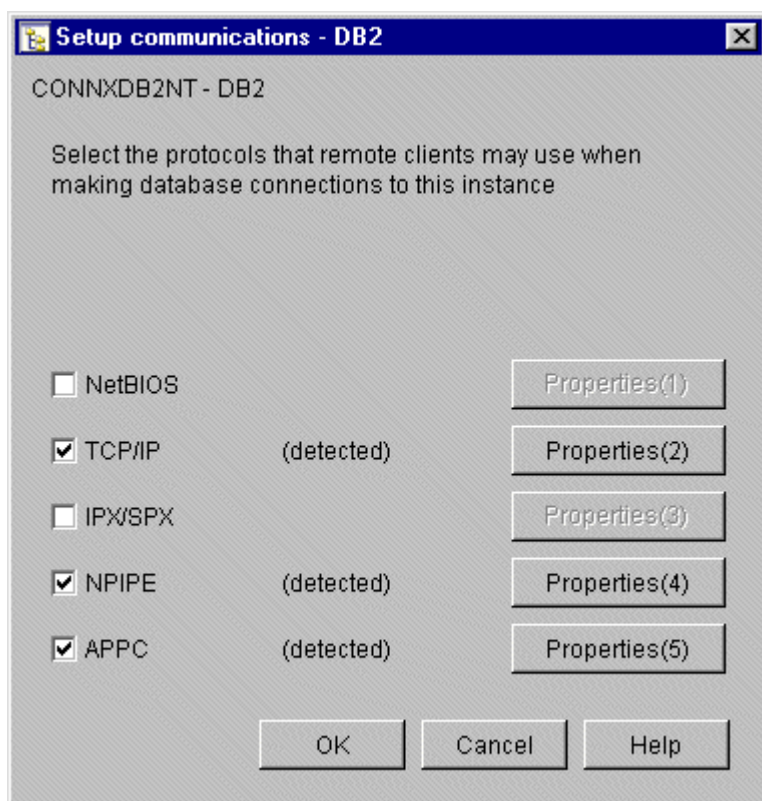
DB2/UDB-APPC/LU: To verify the server APPC/LU 6.2 setting via the DB2 UDB Control Center

Important: The detected configuration is for an IBM Communications Server Version 6.0.2 installed on the same Windows Server with DB2 UDB 6.1. The IBM Communications Server is configured to support APPC / LU 6.2 connectivity via Anynet/MPTN (Multi-Protocol Transport Networking). To connect to this DB2 UDB server via the IBM Communications server software, you need to have the IBM Communications Server client installed on your machine, or an SNA vendor client which supports AnyNet/MPTN.

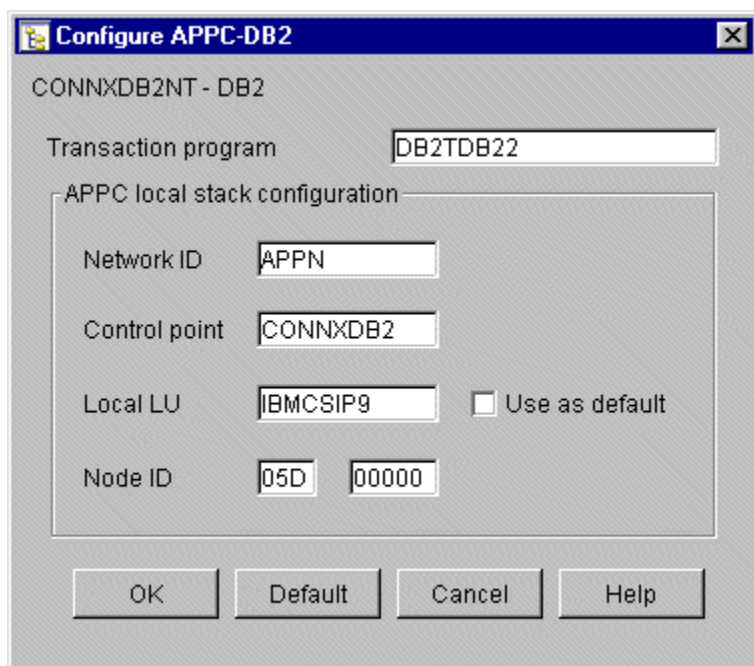
1. Start the **DB2 Control Center**. Click on a DB2 instance in the left pane, right-click and select **Setup communications**.



- If APPC is pre-configured, the **APPC** check box is selected.



- Click the **Properties** button to the right of the APPC check box in the **Setup Communications** dialog box. The **Configure APPC-DB2** dialog box appears.



To verify server APPC/LU 6.2 settings via Command Line Processor

1. At the target server, start the DB2 Command Line Processor window and enter the following command:
Get DBM CFG
2. Verify that a line such as the following appears:
APPC (TPNAME) DB2TDB22
3. Note that this entry matches the value displayed in the APPC-DB2 dialog box.

To add APPC/LU 6.2 support to the configuration

- Open the DB2 Command Line Processor window. Type the following command:
UPDATE DBM CFG USING TPNAME tpname

where **tpname** is a 1- to 64-character APPC transaction program name.

This transaction program name must also be defined in the SNA vendor server configuration.

CONNX and DB2/UDB - AIX Servers

To verify server settings for AIX Servers

1. The /etc/services file must contain connect and interrupt port definitions for DB2, for example:

```
db2cdb2  50000/tcp    # Connection port for DB2 instance db2aix
db2idb2  50001/tcp    # Interrupt port for DB2 instance db2aix
```

The db2idb2 port value is always 1 greater than the db2cdb2 port number.

The value of the DB2 connection port service name varies per DB2 UDB AIX installation, but it must match the corresponding SVCENAME configuration variable in step 3.

2. Using the DB2 administrator account, log on to the DB2 UDB AIX server and issue the following command:

```
$ db2set -all
```

The following environment variable/value is necessary to support TCP/IP connectivity from CONNX to DB2 UDB:

```
DB2COMM=TCPIP
```

If the DB2COMM environment variable is not set, update it as follows:

```
$ db2set DB2COMM=TCPIP
```

3. Next, enter the following:

```
$ db2 get dbm cfg
```

Verify that the following configuration variables are set:

```
Database manager authentication
AUTHENTICATION) = SERVER
TCP/IP Service name  SVCENAME) = db2cdb2
```

If the AUTHENTICATION or SVCENAME variables are null or incorrect, update them as follows:

```
$ db2 update dbm cfg using authentication server
$ db2 update dbm cfg using svcename db2cdb2
```

4. Finally, if the any updates were made to the DB2 configuration or environment variables, stop and restart the DB2 instance with the following commands:

```
$ db2stop
$ db2start
```


CONNX and DB2/400 Databases

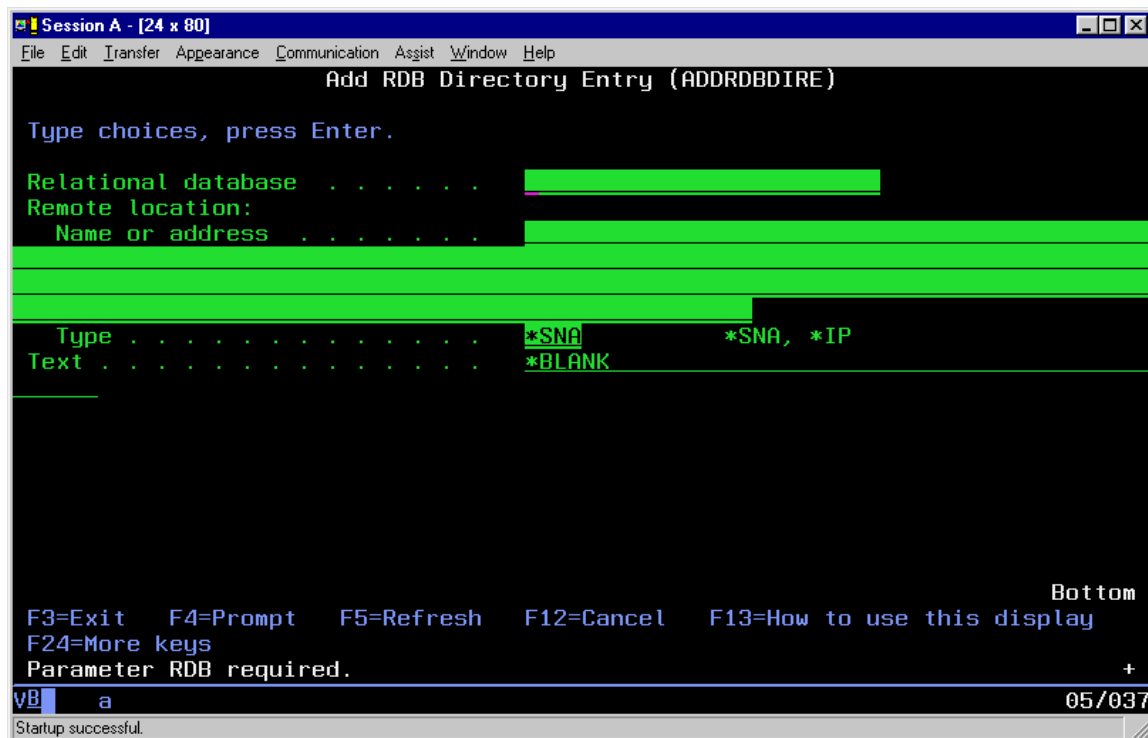
Communications Setup and Scenarios for TCP/IP

To establish RDBName configuration

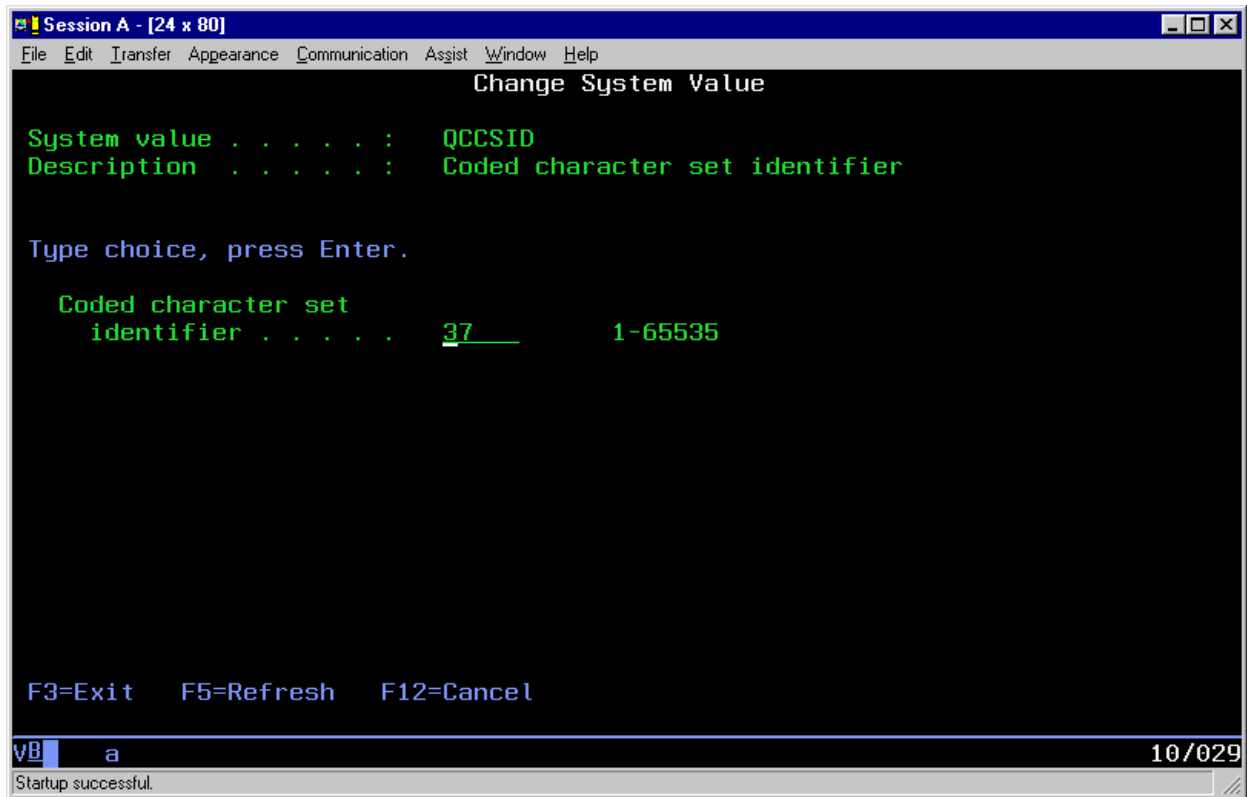
1. Start a 5250 terminal emulator session and log on to the target OS/400. Verify the host RDBNAME by entering the following command at the command line prompt:

DSPRDBDIRE

If a remote location of type *LOCAL does not exist, define it via the ADDRDBDIRE (Add Rdb Directory Entry) command.



2. Verify and/or change the single-byte CCSID (Coded Character Set Identifier) from the default of 65535 to 37 via the WRKSYSVAL (Work with System Values) command.



Important: All OS/400 systems ship with the single-byte QCCSID system value set to 65535, which is an undefined code page. Leaving the default QCCSID system value at 65535 can cause DB2/400 CHAR, VARCHAR, and LONGVARCHAR columns to appear as BINARY columns to CONNX for DB2. You can choose not to change this system value, but then all character columns will appear as binary data, unless you check the Convert Binary to Character check box when importing/defining a CONNX for DB2/400 CDD in the Import CDD dialog box. Refer to the Settings tab in the Import CDD dialog box.

To establish TCP/IP configuration

Ask your system administrator for the symbolic or 4-part dotted numeric TCP/IP address.

1. Start a 5250 terminal emulator session, log on to the target OS/400, and type
`go tcpadm`
at the main command line.
2. The Work with TCP/IP Network Status window appears. Type the number **7** on the main command line.
3. Under **Opt**, type the number **3**, and then verify that an entry for DRDA appears under Local Port.

```

Session A - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help

Work with TCP/IP Connection Status

Local internet address . . . . . : *ALL
System:

Type options, press Enter.
4=End 5=Display details

Opt Remote Address Remote Port Local Port Idle Time State
--
* * * ftp-con > 448:34:41 Listen
* * * telnet 024:42:07 Listen
* * * smtp 448:34:41 Listen
* * * pop3 863:10:23 Listen
* * * snmp 006:29:07 *UDP
* * * APPCove > 006:47:49 Listen
* * * APPCove > 000:00:01 *UDP
* * * drda 001:01:08 Listen
* * * as-svrmap 744:41:55 Listen
* * * exec 863:10:57 Listen
* * * lpd 863:11:00 Listen

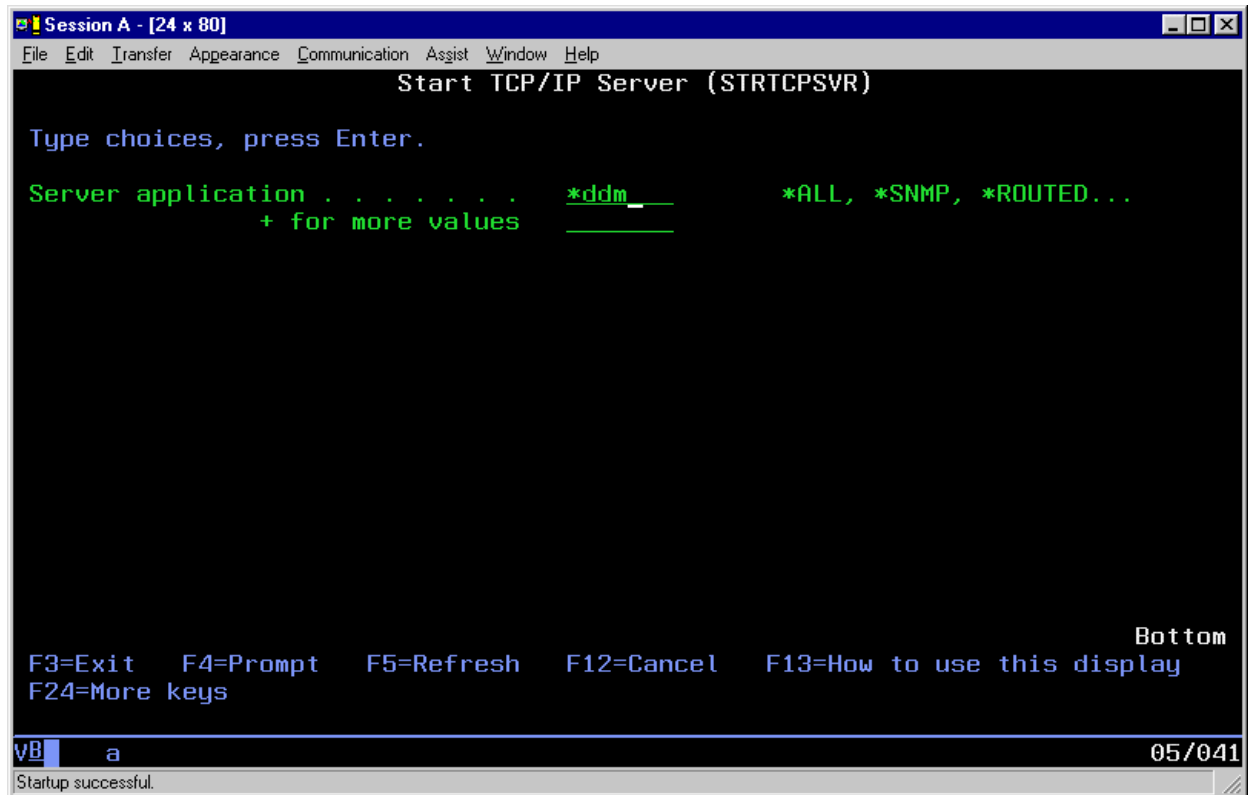
More...

F5=Refresh F11=Display byte counts F13=Sort by column
F14=Display port numbers F22=Display entire field F24=More keys

VB a 10/003
Startup successful.

```

4. If the DRDA port entry does not exist, go to the TCPADM menu and select **option 5= Start TCP/IP Servers**.
5. On the next screen, type ***DDM** in the server application field.

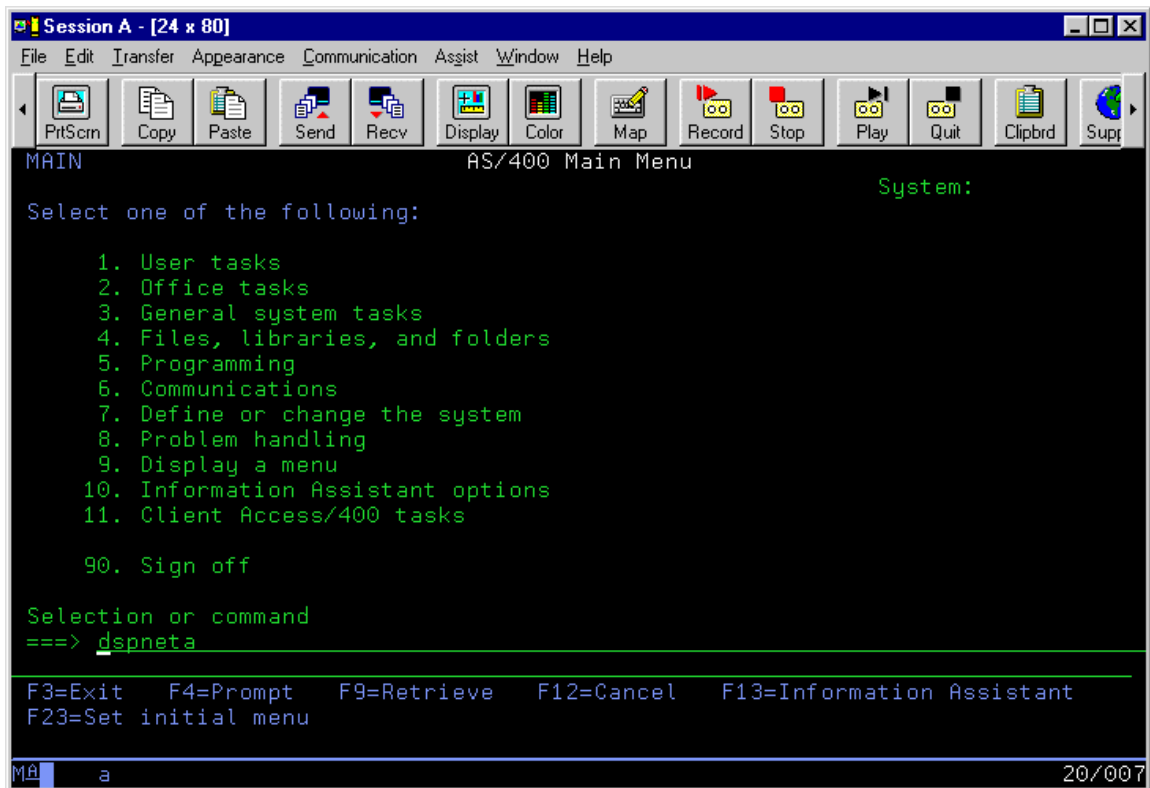


Setting up APPC/LU 6.2 Communications on Target Systems

To set up MPTN configuration

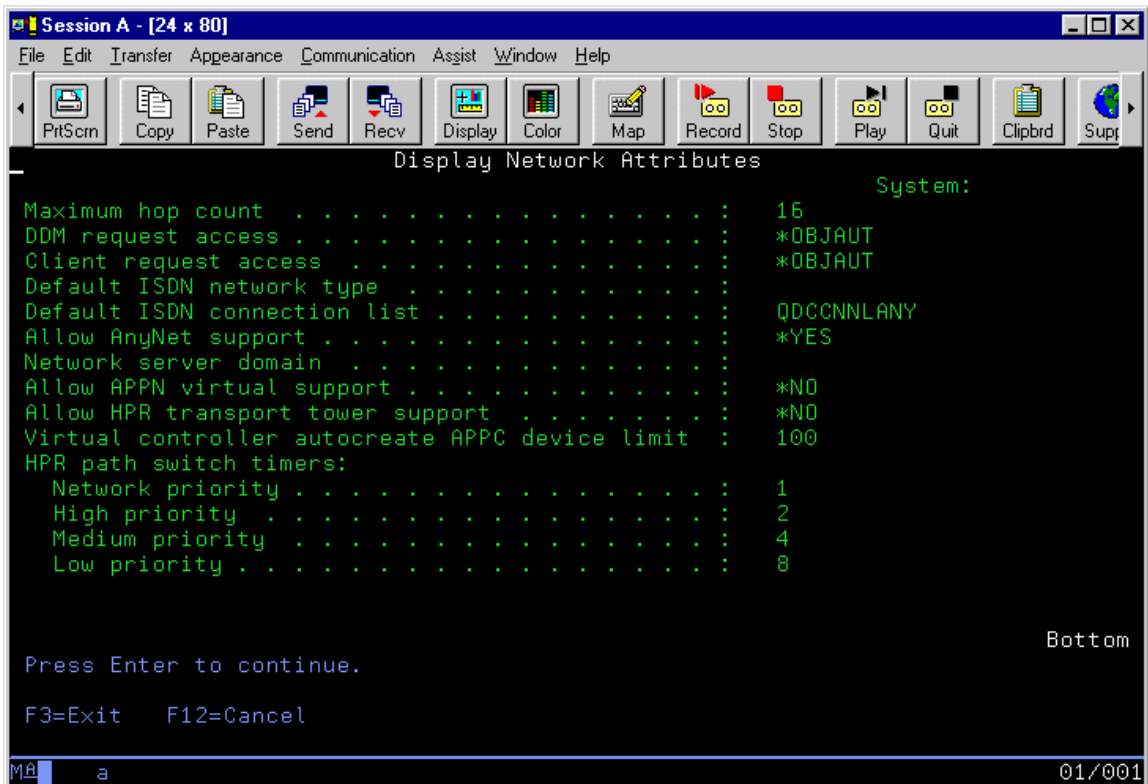
This section assumes that your AS/400 is already configured to support TCP/IP.

1. Enter the **Display Network Attributes (DSPNETA)** command on the main command line in the AS/400 main menu.



This command displays the network attributes configured for your host system.

2. Page down and verify that the **Allow AnyNet Support** parameter is set to ***YES**.



```

Session A - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Supr

Display Network Attributes

Maximum hop count . . . . . : 16
DDM request access . . . . . : *OBJAUT
Client request access . . . . . : *OBJAUT
Default ISDN network type . . . . . :
Default ISDN connection list . . . . . : QDCCNNLANY
Allow AnyNet support . . . . . : *YES
Network server domain . . . . . :
Allow APPN virtual support . . . . . : *NO
Allow HPR transport tower support . . . . . : *NO
Virtual controller autcreate APPC device limit : 100
HPR path switch timers:
  Network priority . . . . . : 1
  High priority . . . . . : 2
  Medium priority . . . . . : 4
  Low priority . . . . . : 8

Press Enter to continue.
F3=Exit F12=Cancel

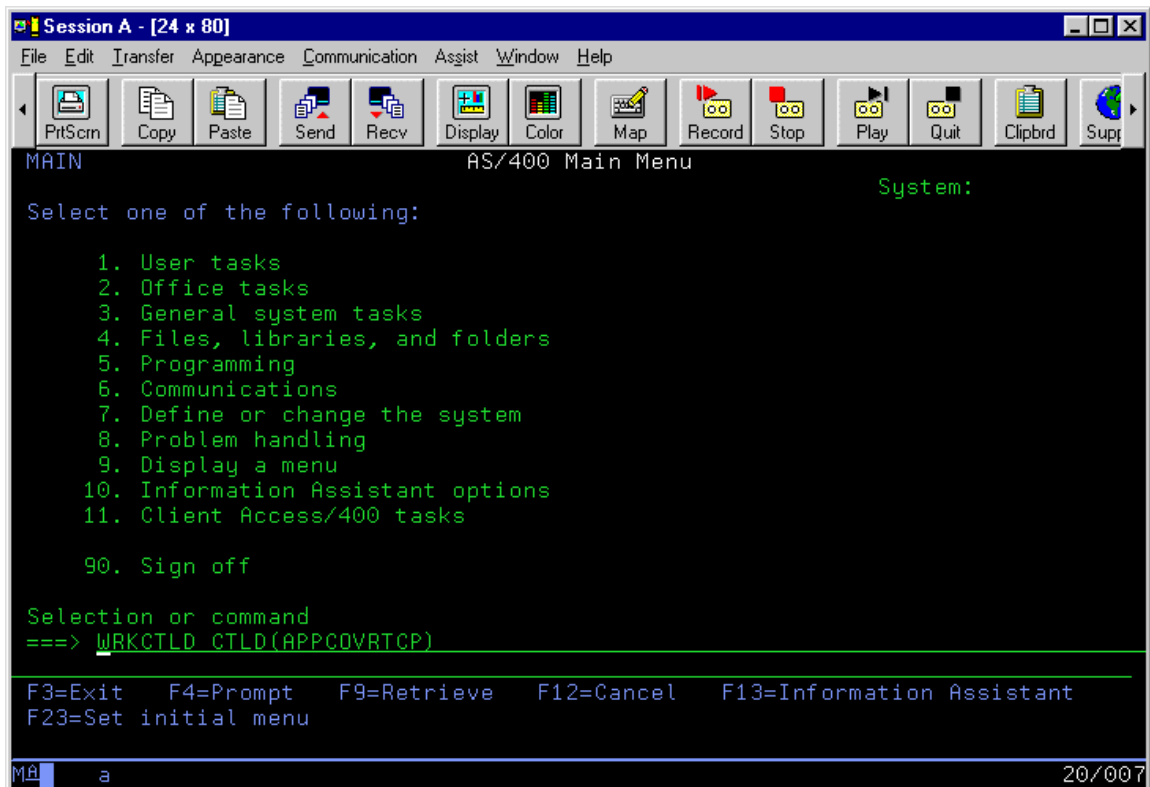
Bottom

01/001

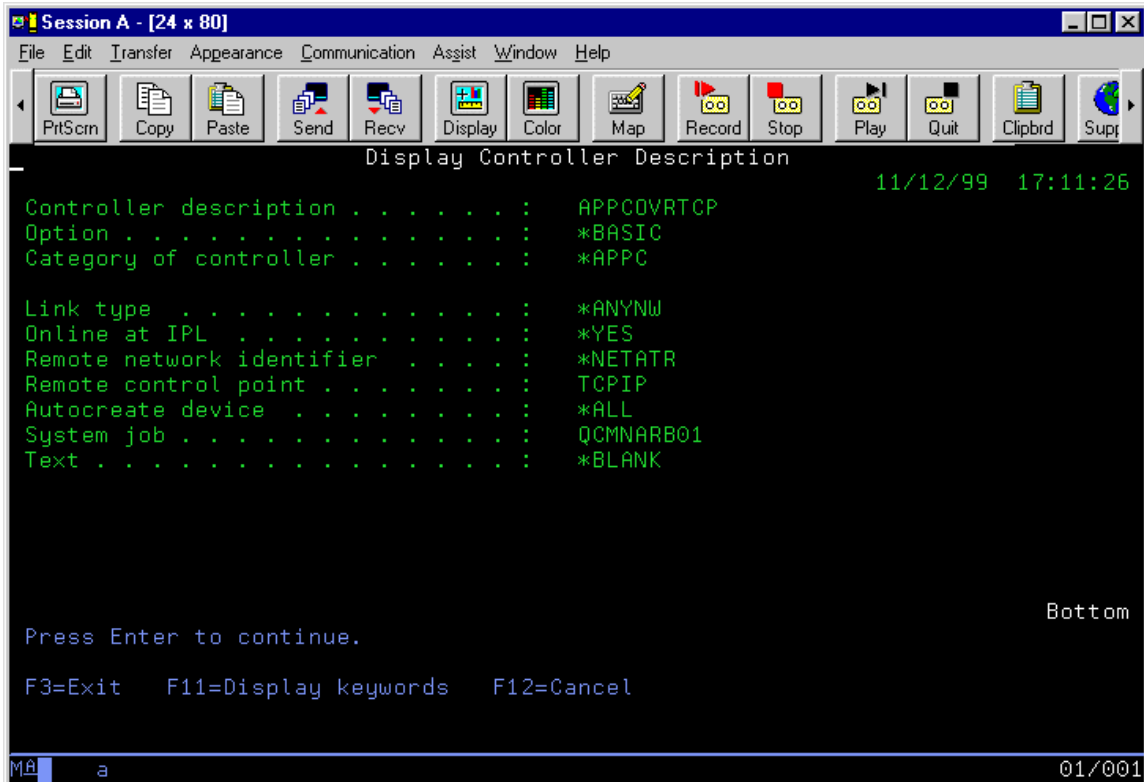
```

To verify MPTN APPC Controller configuration

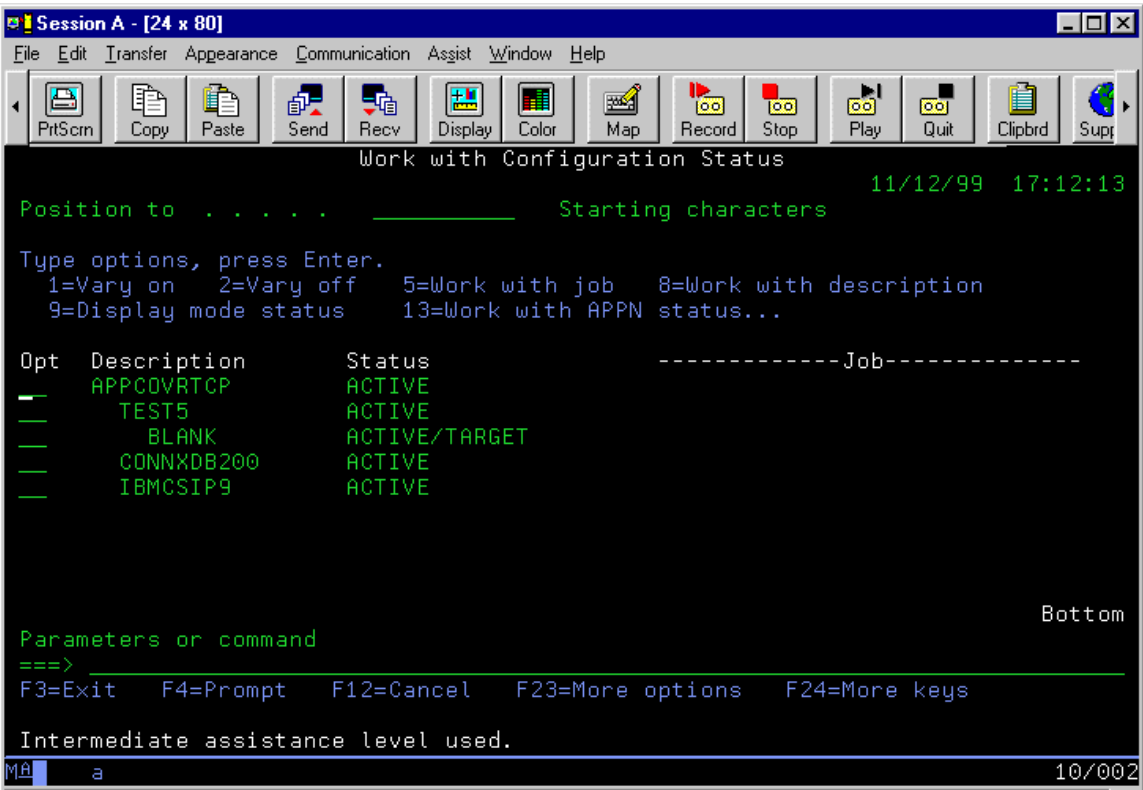
1. Use the **WRKCTLD** command to verify that the MPTN controller is defined and active.



2. Page down and verify that the Category of Controller is set to APPC.



3. Verify that the **APPCOVRTCP** option is set to **Active** in the **Work with Configuration Status** window.



To verify IEEE 802.2 DLC configuration

1. Use the **WRKLIND** command to open the Work with Line Description window.

In this example, line description ETHSNA has been defined to support SNA via Ethernet/DLC. The important parameters are the local adapter address and the exchange identifier.

Make a note of these; they are used in the configuration of three-tier server products such as IBM Communications Server, Microsoft SNA Server, and Novell NetWare for SAA; and in two-tier products such as Attachmate Extra! PC or NetManage RUMBA.

2. The Display Line Description window appears. Verify the Local Adapter Address of the OS/400 and the Exchange Identifier.

```

Session A - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScr Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Supr
Display Line Description
11/12/99 17:26:39
Line description . . . . . : ETHSNA
Option . . . . . : *BASIC
Category of line . . . . . : *ELAN
Resource name . . . . . : CMN02
Online at IPL . . . . . : *YES
Vary on wait . . . . . : *NOWAIT
Local adapter address . . . . . : 020000000001
Exchange identifier . . . . . : 05600000
Ethernet standard . . . . . : *ALL
Line speed . . . . . : 10M
Current line speed . . . . . : 10M
Duplex . . . . . : *HALF
Current duplex . . . . . : *HALF
Maximum controllers . . . . . : 40
Error threshold level . . . . . : *OFF
Generate test frame . . . . . : *YES
More...
Press Enter to continue.
F3=Exit F11=Display keywords F12=Cancel
Mâ a 01/001

```

To verify IBMRDB mode description

1. Start a 5250 terminal emulator session, and then log on to the target OS/400.
2. Verify the IBMRDB mode description with the DSPMODD command. The default mode name for APPC/DRDA conversations is IBMRDB; it is predefined on all OS/400 machines.
3. Verify that the mode name exists.

Note the parameters for maximum sessions, maximum conversations, locally controlled sessions, and inbound/outbound pacing values.

```

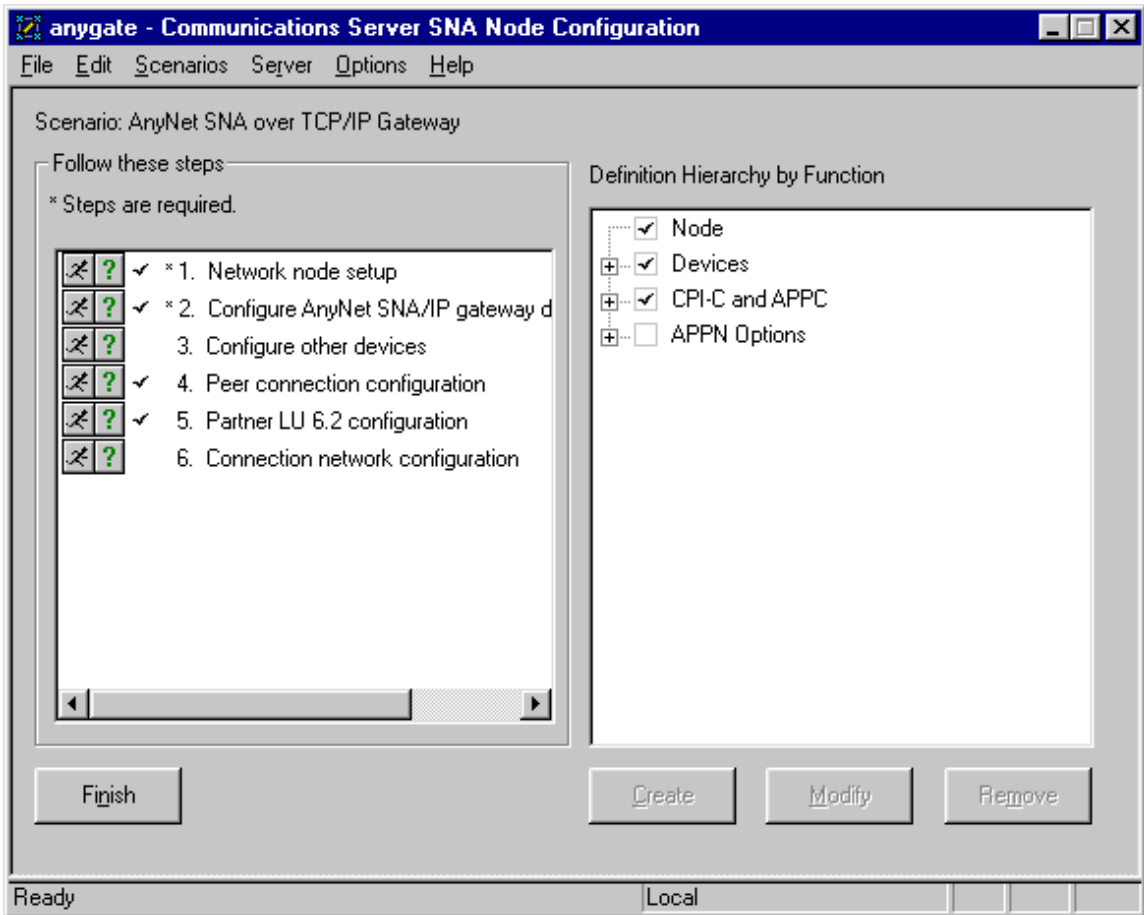
Session A - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Supr
Display Mode Description
11/12/99 16:57:31
Mode description . . . . . : IBMRDB
Class-of-service . . . . . : #BATCH
Maximum sessions . . . . . : 8
Maximum conversations . . . . . : 8
Locally controlled sessions . . . . . : 4
Pre-established sessions . . . . . : 0
Maximum inbound pacing value . . . . . : *CALC
Inbound pacing value . . . . . : 3
Outbound pacing value . . . . . : 3
Maximum length of request unit . . . . . : *CALC
Data compression . . . . . : *NETATR
Inbound data compression . . . . . : *RLE
Outbound data compression . . . . . : *RLE
Session level encryption . . . . . : *NONE
Text . . . . . : This Mode is IBM Supplied
Bottom
Press Enter to continue.
F3=Exit F11=Display keywords F12=Cancel
MA a 01/001

```

Communications Setup and Scenarios for 3-Tier Communications Server

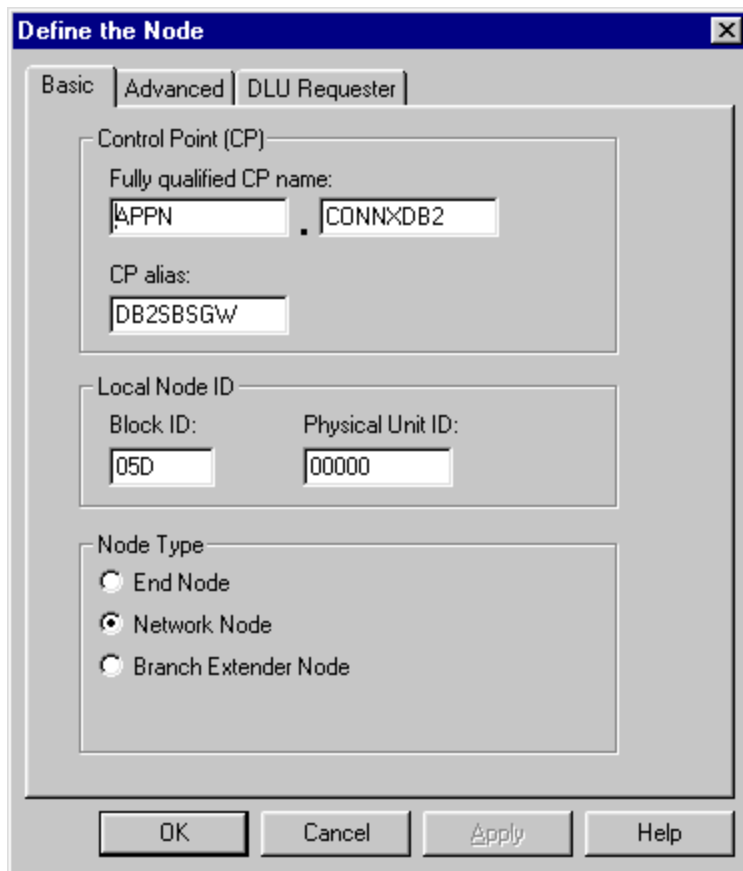
To set up the IBM Com Server as an MPTN gateway

1. Select **Start**, point to **Programs**, and then select the **IBM Communications Server SNA Node Configuration** application.
2. Select **Scenarios** on the menu bar, and then select **AnyNet SNA over TCP/IP Gateway**.



3. The left pane of this scenario window lists the following steps. Double-click Step 1. The **Define the Node** dialog box appears.
4. On the **Basic** tab, define the network node setup. Type the fully qualified CP name and the CP alias under **Control Point**. In this sample, APPN.CONNXDB2 is entered as the fully qualified CP name, and is shown with the LU alias DB2SBSGW. Accept the defaults for Block ID and Physical Unit ID.

Under **Node Type**, select **Network Node**. The **Advanced** and **DLU Requester** tabs can be left with their current default values for this application.

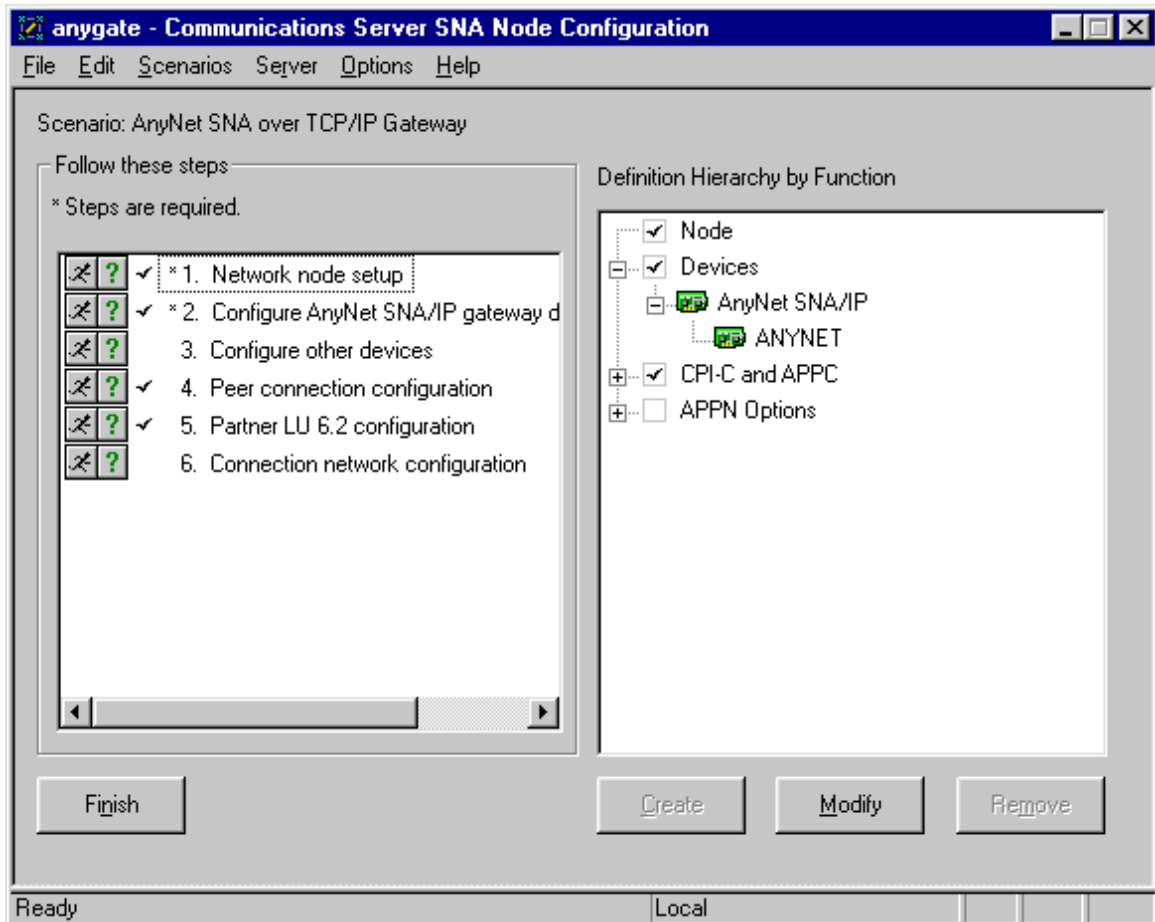


The image shows a Windows-style dialog box titled "Define the Node". It has three tabs: "Basic", "Advanced", and "DLU Requester". The "Basic" tab is selected. Inside the dialog, there are three main sections:

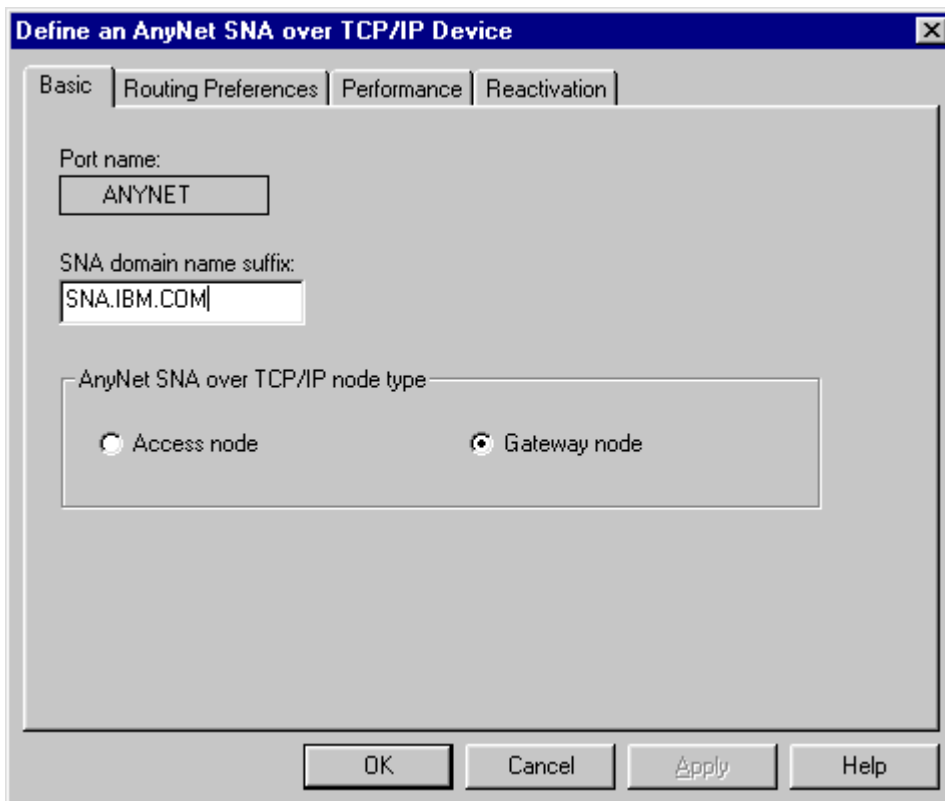
- Control Point (CP):** This section contains a label "Fully qualified CP name:" followed by two text boxes. The first box contains "APPN" and the second box contains "CONNxDB2". Below this is a label "CP alias:" followed by a text box containing "DB2SBSGW".
- Local Node ID:** This section contains two labels: "Block ID:" and "Physical Unit ID:". Below "Block ID:" is a text box containing "05D". Below "Physical Unit ID:" is a text box containing "00000".
- Node Type:** This section contains three radio buttons:
 - ☐ End Node
 - ☒ Network Node
 - ☐ Branch Extender Node

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

5. Click the **OK** button to return to the **Communications Server SNA Node Configuration** dialog box. The node setup appears in the right pane.
6. Double-click Step 2, Configure SNA/IP gateway device. The AnyNet SNA/IP device is auto-configured when the AnyNet gateway scenario is selected. Double-click on the ANYNET device name in the right pane to show its properties.



7. The Define an AnyNet SNA over TCP/IP Device dialog box appears.



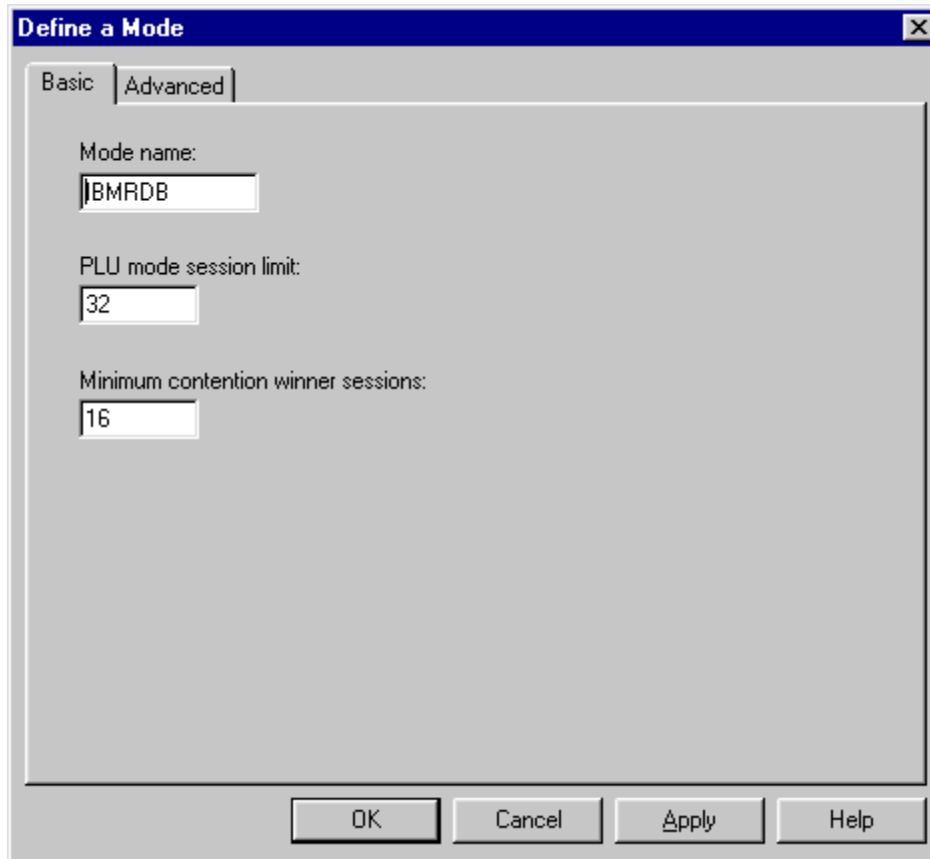
8. Select the **Basic** tab to view the port name and the default SNA domain name suffix. Under **AnyNet SNA over TCP/IP** node type, verify that **Gateway** node is selected.
9. The SNA domain name suffix is concatenated onto the target luname.netid to map the dotted TCP/IP address to a netid.luname = APPN.CONNXDB2. To implement this mapping, the Windows server needs the following entries in its winnt\system32\drivers\etc\hosts file:


```
012.345.678.901 connxdb2 #=localhost
012.345.678.901 connxdb2.appn.sna.ibm.com #=localhost
```

 where 012.345.678.901 is an example dotted IP address for the server machine. This entry need not be present in the hosts file if a symbolic entry is defined in your network domain name server.
10. Click the **OK** button to return to the Communications Server SNA Node Configuration dialog box.

To define the mode name for DRDA connections = IBMRDB

1. IBM has defined IBMRDB as a special mode for use by APPC / DRDA connections. To define the mode name from within the Definition Hierarchy by Function pane in the Communications Server SNA Node Configuration dialog box, verify that **Step 2, Configure AnyNet SNA/IP Gateway Device** is selected. Select **Modes** under the **CPI-C** and **APPC** directory folder in the right pane, and then click the **Create** button.
2. The **Define a Mode** dialog box appears.



The image shows a 'Define a Mode' dialog box with a blue title bar and a close button. It has two tabs: 'Basic' (selected) and 'Advanced'. The 'Basic' tab contains three text input fields: 'Mode name:' with 'IBMRDB' entered, 'PLU mode session limit:' with '32' entered, and 'Minimum contention winner sessions:' with '16' entered. At the bottom, there are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

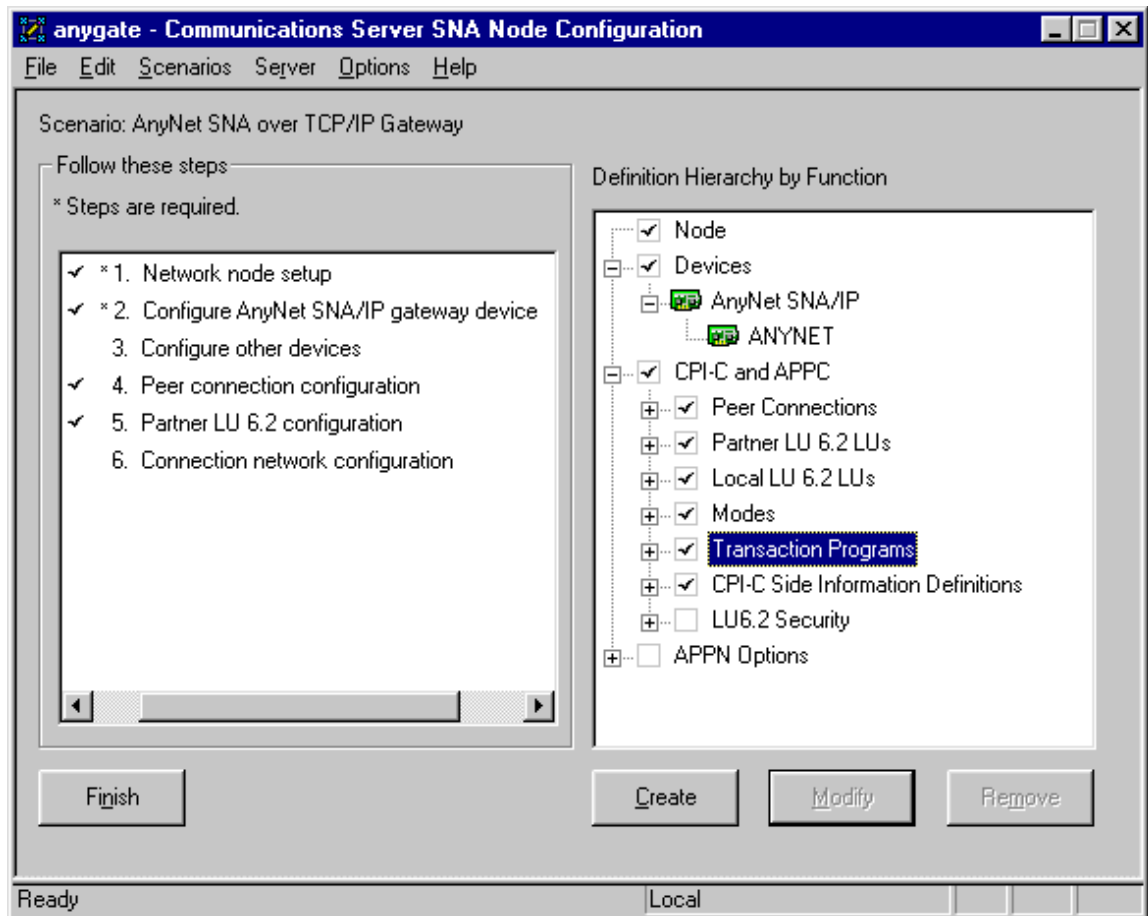
3. On the **Basic** tab, enter **IBMRDB** as the mode name. The values entered for PLU mode session limit and minimum contention winner sessions should match the values defined at the target host.
4. Click the **OK** button to return to the **Communications Server SNA Mode Configuration** dialog box.

Related Topic

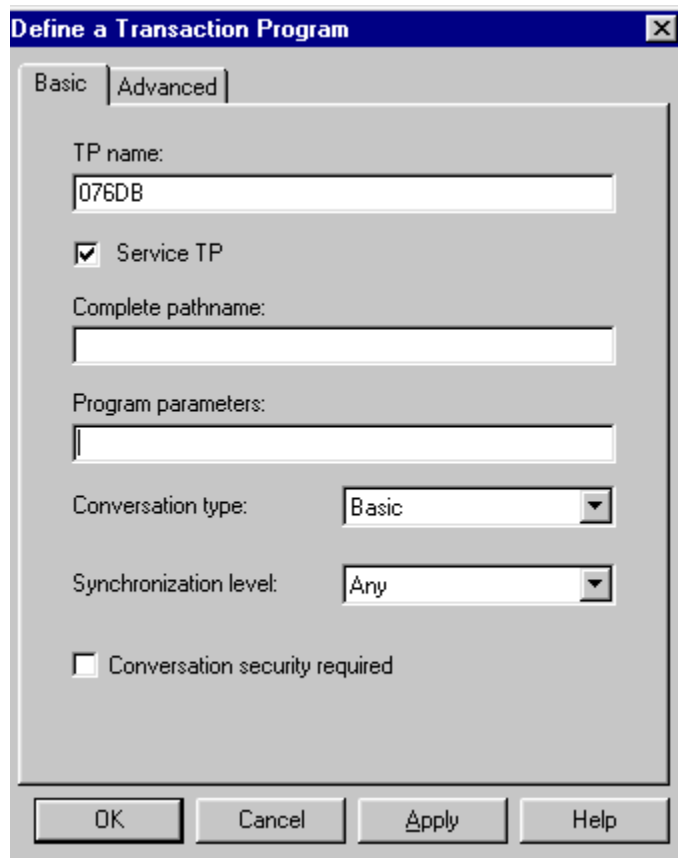
[➤ To verify IBMRDB mode description](#)

To define or verify the DB2 Transaction Program Name

1. Select **Transaction Program** in the **CPI-C** and **APPC** directory folder in the **Communications Server SNA Node Configuration** dialog box, and then click the **Create** button.



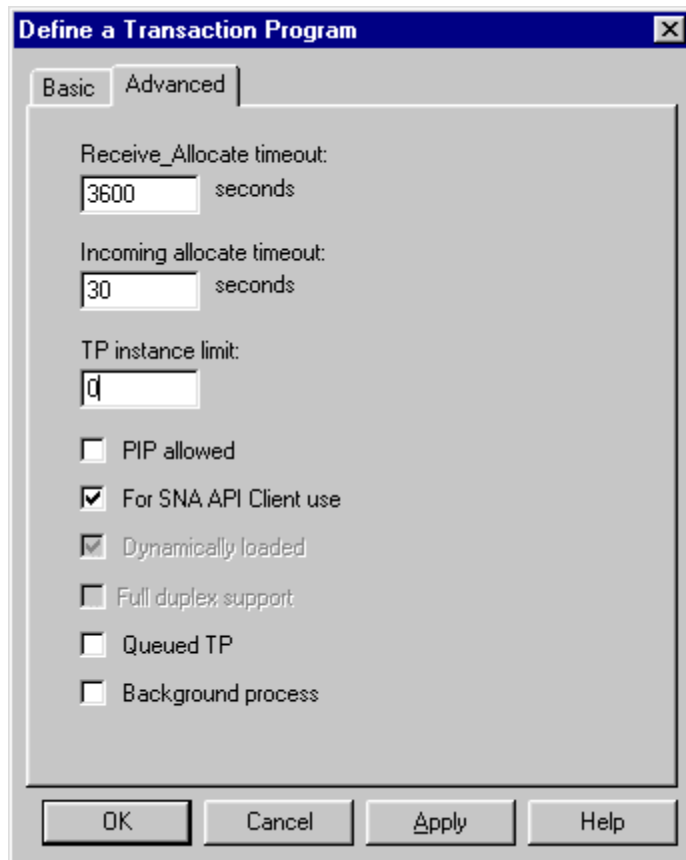
2. The list of defined transaction programs appears. The default APPC transaction program for DRDA is 0x07F6C2C4 = .6DB. Define this program to your IBM Communications Server configuration as follows:



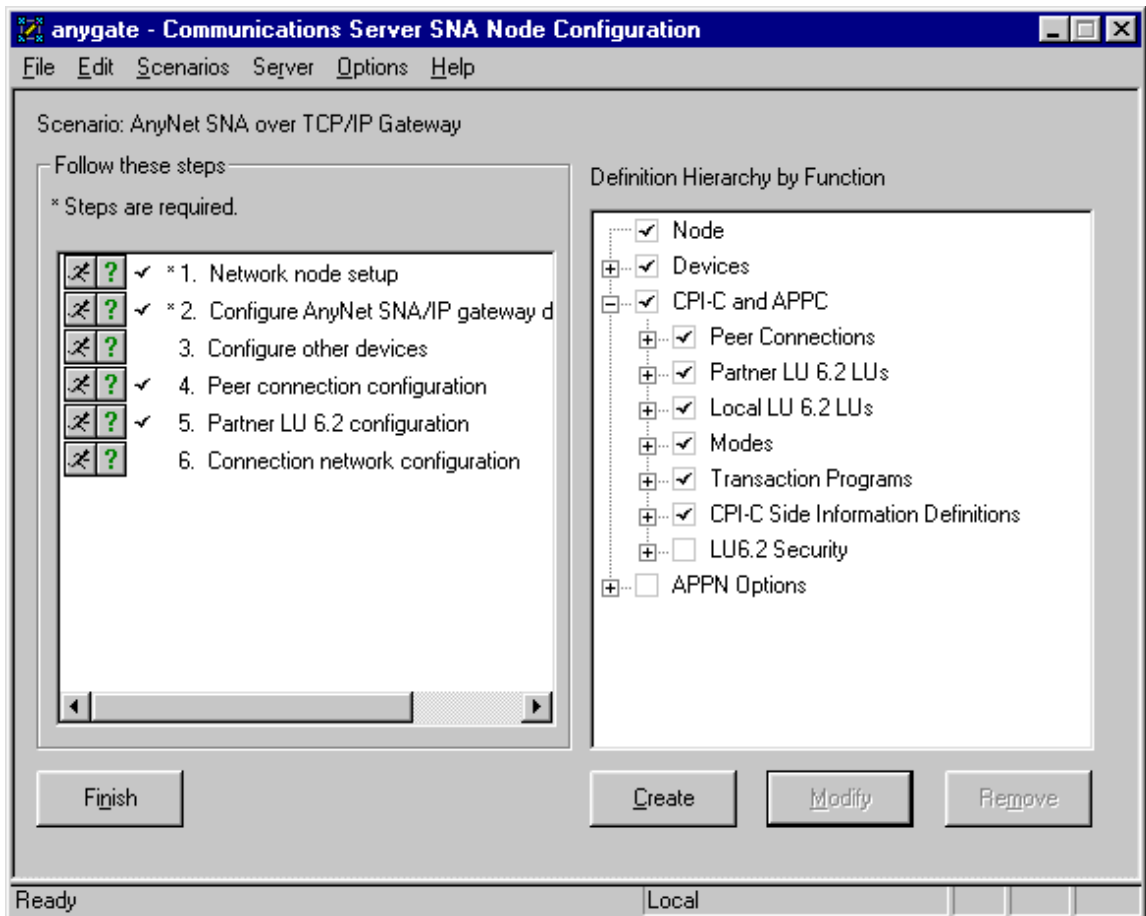
The image shows a Windows-style dialog box titled "Define a Transaction Program". It has two tabs: "Basic" (selected) and "Advanced". The "Basic" tab contains the following fields and controls:

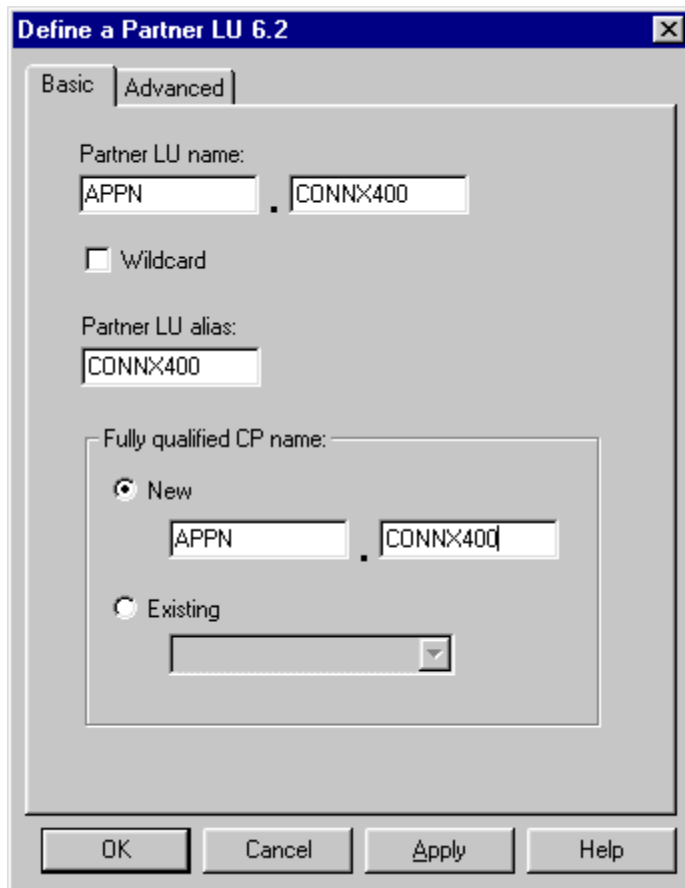
- TP name:** A text box containing "076DB".
- Service TP:** A checked checkbox.
- Complete pathname:** An empty text box.
- Program parameters:** An empty text box.
- Conversation type:** A dropdown menu with "Basic" selected.
- Synchronization level:** A dropdown menu with "Any" selected.
- Conversation security required:** An unchecked checkbox.

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".



3. From the **Definition Hierarchy By Function** pane, select **Partner LU 6.2 LU's** and click the **Create** button.





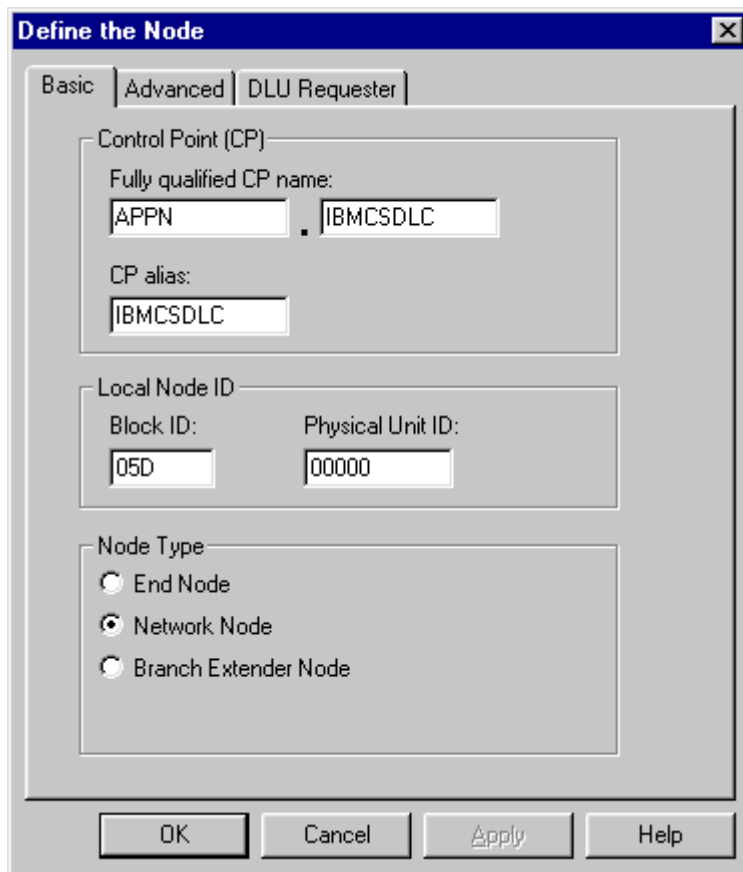
4. The MPTN mapping from TCP/IP address to an APPC netid.luname is implemented in the **\winnt\system32\drivers\etc\hosts** file, as follows:

```
012.345.678.012 connx400.appn.sna.ibm.com #Anynet
```

5. Thus, the SNA netid.luname is appn.connx400, which should match the Local Network ID and local control point name fields as displayed in the OS/400 DSPNETA command. The sna.ibm.com suffix is the default SNA domain name suffix defined by the IBM Communications Server AnyNet Gateway Scenario as shown in [To set up the IBM Com Server as an MPTN Gateway](#). If this entry is not present in your Windows server hosts file, it must exist in a TCP/IP domain name server on your network. The dotted IP address is the address of the target OS/400 machine.

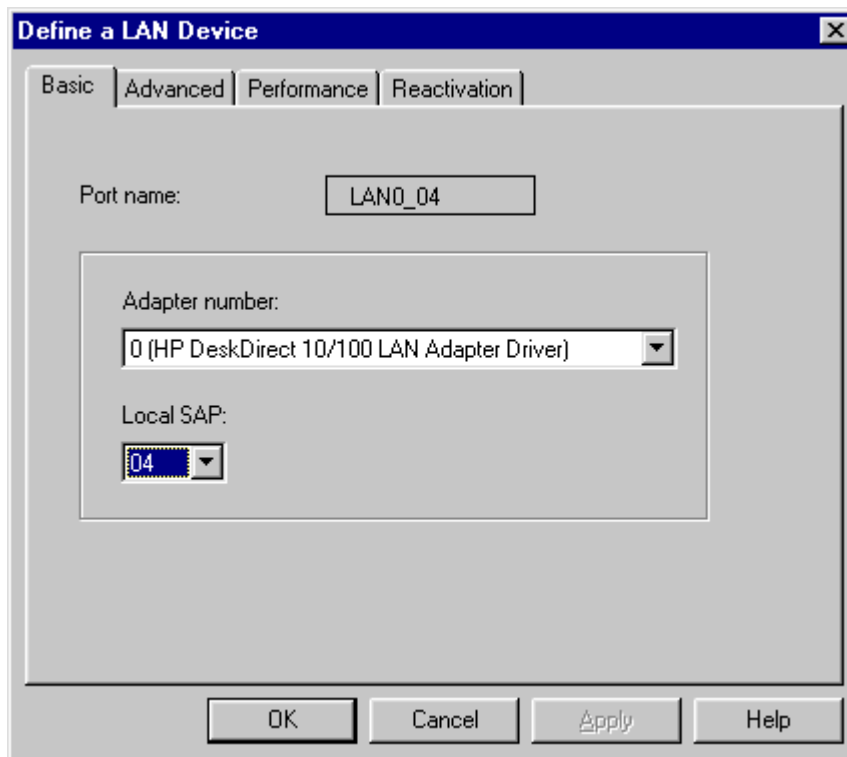
To set up the IBM Communications Server in an IEEE 802.2 DLC scenario

1. Select **Start**, point to **Programs**, and select the **IBM Communications Server SNA Node Configuration** application. Select **Scenarios** on the main menu. The **Choose a Configuration** dialog box displays. Select **APPN Network Node**, and then click the **Finish** button to return to the Communications Server SNA Node Configuration window.
2. Double-click **Network Node Setup** in the left pane. The **Define The Node** dialog box appears. For this example, the netid is defined as APPN and the server node is defined as IBMCS DLC under **CP Name** and **CP Alias**. Note that the **Network Node** option is selected under **Node Type**. Accept the defaults for Block ID and Physical Unit ID, and for the Advanced and DLU Requester tabs. Click the **OK** button to return to the Communications Server SNA Node Configuration window.

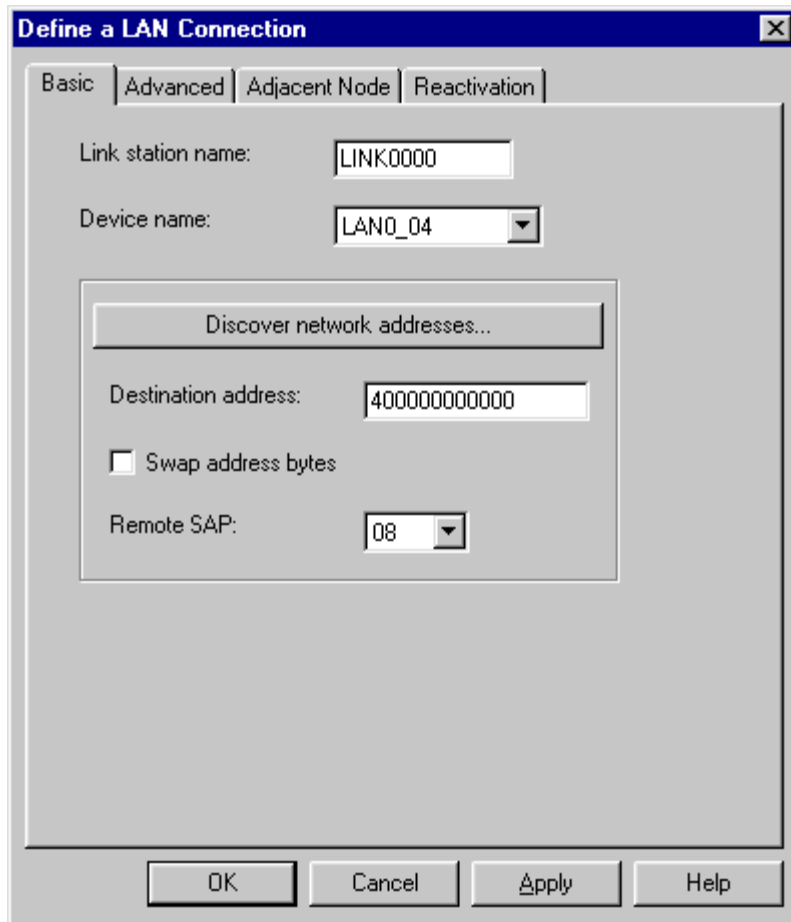


The image shows the 'Define the Node' dialog box with the 'Basic' tab selected. The 'Control Point (CP)' section contains 'Fully qualified CP name:' with two text boxes: 'APPN' and 'IBMCS DLC', separated by a period. Below this is 'CP alias:' with a text box containing 'IBMCS DLC'. The 'Local Node ID' section has 'Block ID:' with a text box containing '05D' and 'Physical Unit ID:' with a text box containing '00000'. The 'Node Type' section has three radio buttons: 'End Node' (unselected), 'Network Node' (selected), and 'Branch Extender Node' (unselected). At the bottom are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

3. Double-click **Device Configuration** in the left pane. Select **Devices** in the **Definition Hierarchy by Function** pane, and then click the **Create** button. The **Device Type** dialog box appears. Choose **LAN** in the list box; and then click the **OK** button to return to the Communications Server SNA Node Configuration window.
4. Expand the **LAN** folder in the right pane. The **Define a LAN Device** dialog box appears. Select or accept the current LAN Adapter described on the **Basic** tab. Accept the defaults displayed on the **Advanced**, **Performance**, and **Reactivation** tabs, and then click the **OK** button to return to the Communications Server SNA Node Configuration window.



5. Double-click **Peer Connection Configuration** in the left pane. Expand the **CPI_C** and **APPC** and **Peer Connections** nodes in the right pane. Select **Peer Connections**, and then click the **Create** button.
6. The **Define a LAN Connection** dialog box appears. Accept the defaults, and type the LAN adapter (MAC) address of the target OS/400 in the **Destination Address** text box. The LAN adapter address can be displayed via the OS/400 WRKLIND command. If you are not authorized to use this command, ask your OS/400 systems administrator for the target machine LAN adapter (MAC) address.

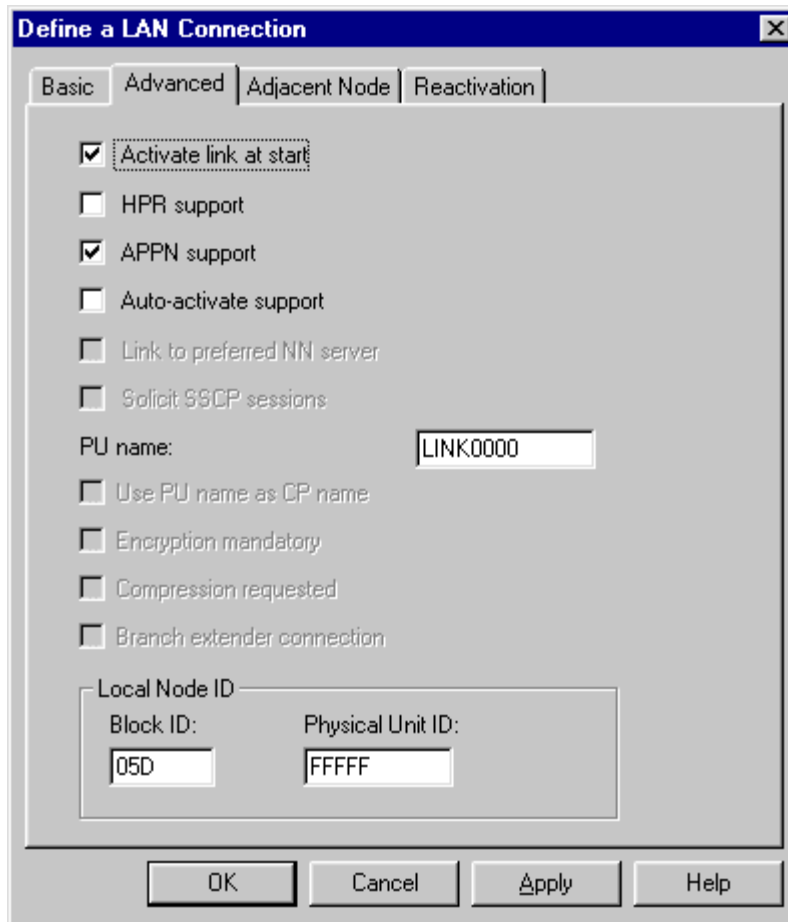


The image shows a Windows-style dialog box titled "Define a LAN Connection". It has four tabs: "Basic", "Advanced", "Adjacent Node", and "Reactivation". The "Basic" tab is selected. Inside the dialog, there are several fields and controls:

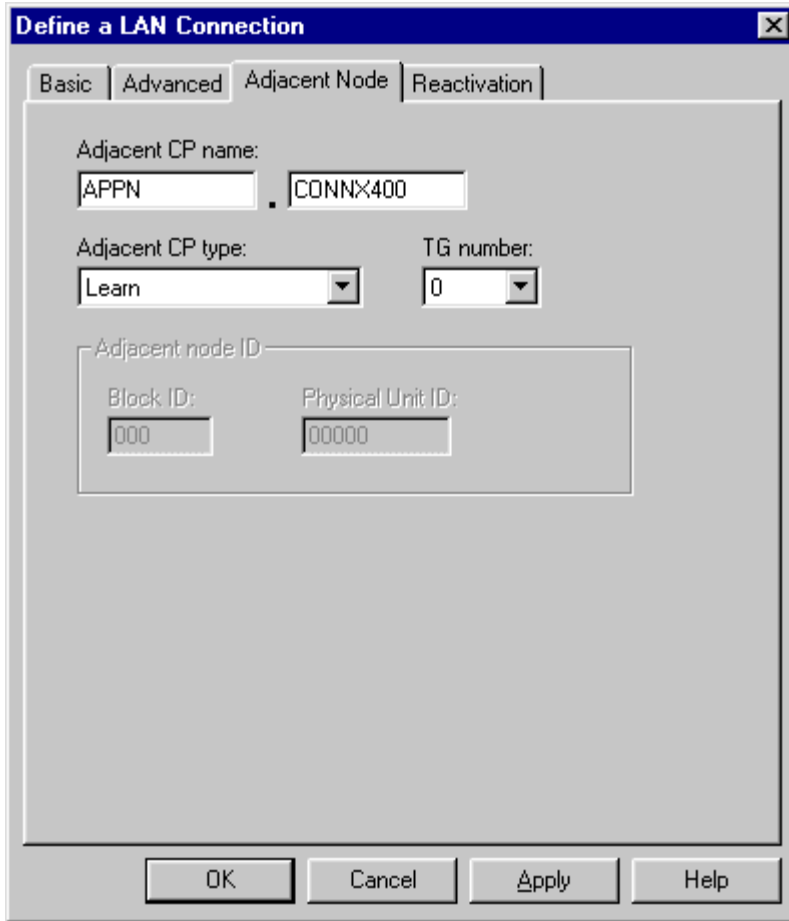
- "Link station name:" with a text box containing "LINK0000".
- "Device name:" with a dropdown menu showing "LAN0_04".
- A button labeled "Discover network addresses..." inside a rectangular frame.
- "Destination address:" with a text box containing "400000000000".
- A checkbox labeled "Swap address bytes" which is currently unchecked.
- "Remote SAP:" with a dropdown menu showing "08".

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

7. Select the **Advanced** tab; check the **APPN Support** check box, and then type a **Block ID = 05D** and a node ID in the range of **00000** to **FFFFF** under **Local Node ID**. Verify that the **Activate Link at Start** and **APPN Support** check boxes are selected.



8. Select the **Adjacent Node** tab; enter the **Adjacent CP Name = APPN.CONNX400** in this example. This value must match the output of the [DSPNETA command](#).

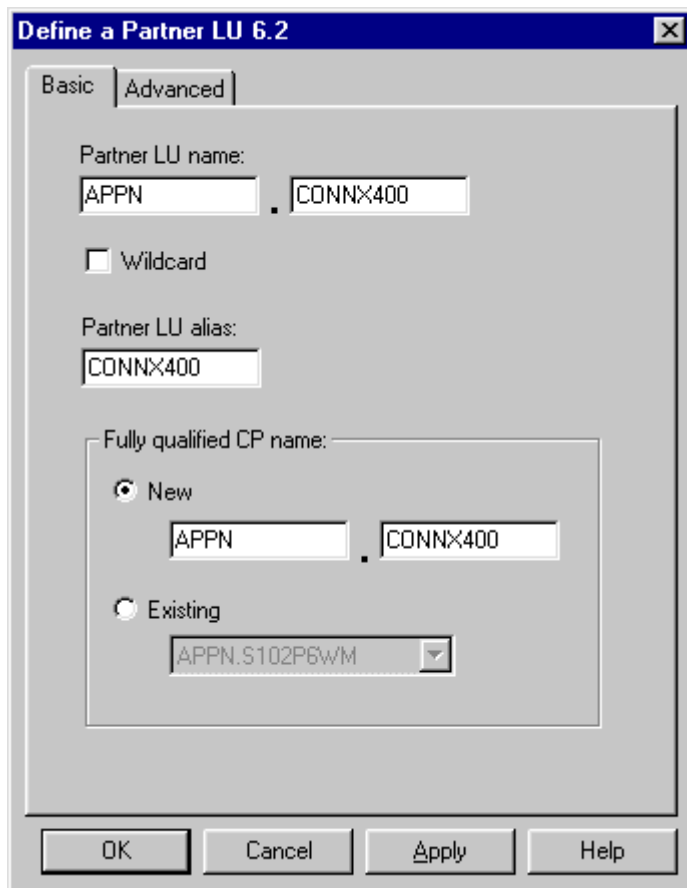


The image shows a Windows-style dialog box titled "Define a LAN Connection". It has four tabs: "Basic", "Advanced", "Adjacent Node" (which is selected), and "Reactivation". The "Adjacent Node" tab contains the following fields:

- Adjacent CP name:** Two text boxes. The first contains "APPN" and the second contains "CONNX400".
- Adjacent CP type:** A dropdown menu showing "Learn".
- TG number:** A dropdown menu showing "0".
- Adjacent node ID:** A group box containing:
 - Block ID:** A text box containing "000".
 - Physical Unit ID:** A text box containing "00000".

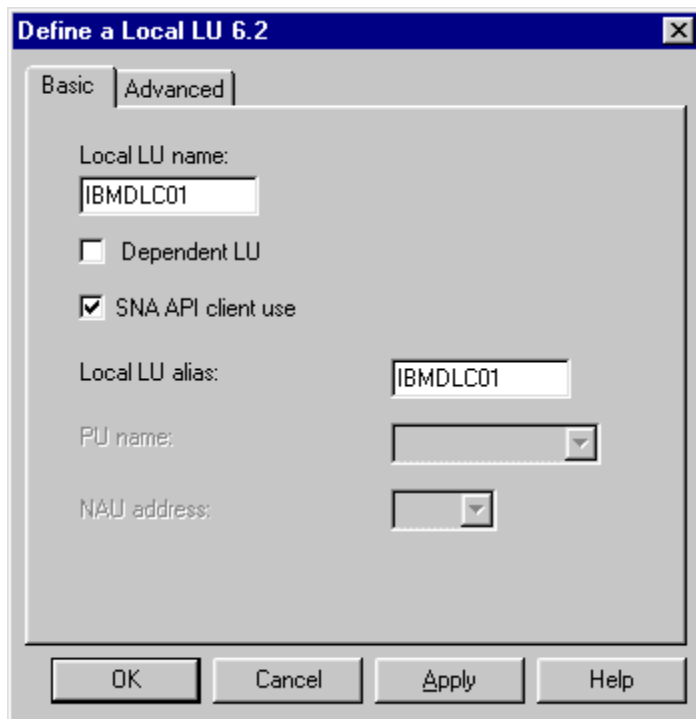
At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

9. Accept the defaults for the **Reactivation** tab (not shown). Click the **OK** button to return to the Communications Server SNA Node Configuration window.
10. Double-click the **Partner LU 6.2 Configuration** in the right pane, and then click the **Create** button. The **Define a Partner LU 6.2** dialog box appears. Type the **Net ID** and the **System name** of the target OS/400, which should match the Local Network ID and Local Control Field Name fields as displayed by the OS/400 [DSPNETA command](#). Also define a partner LU alias and a fully qualified control point (CP) name = netid.system, or in this example, APPN.CONNX400.



The image shows a Windows-style dialog box titled "Define a Partner LU 6.2". It has two tabs: "Basic" and "Advanced". The "Basic" tab is selected. Inside the dialog, there are several input fields and checkboxes. The "Partner LU name:" field contains "APPN" and "CONN400" separated by a dot. Below this is a checkbox labeled "Wildcard" which is unchecked. The "Partner LU alias:" field contains "CONN400". Below that is a section for "Fully qualified CP name:" with two radio buttons: "New" (selected) and "Existing". Under "New", there are two input fields containing "APPN" and "CONN400" separated by a dot. Under "Existing", there is a dropdown menu showing "APPN.S102P6WM". At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

11. View and accept the defaults for the **Advanced** tab (not shown), and then click the **OK** button to return to the Communications Server SNA Node Configuration window.
12. Click the **Local LU 6.2 LUs** check box in the right pane, and then click the **Create** button. The **Define a Local LU 6.2** dialog box appears. Define a local LU and a local LU alias. For simplicity, these are both = IBMDLC01.

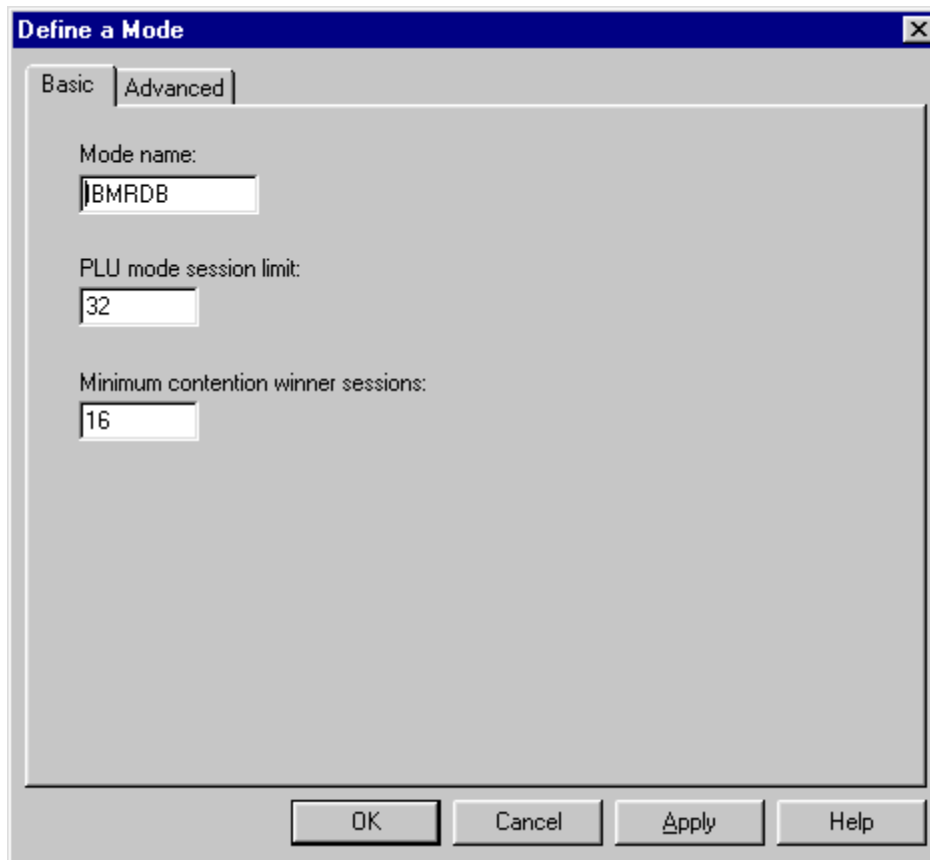


The image shows a Windows-style dialog box titled "Define a Local LU 6.2". It has two tabs: "Basic" and "Advanced". The "Basic" tab is selected. Inside the dialog, there are several fields and checkboxes:

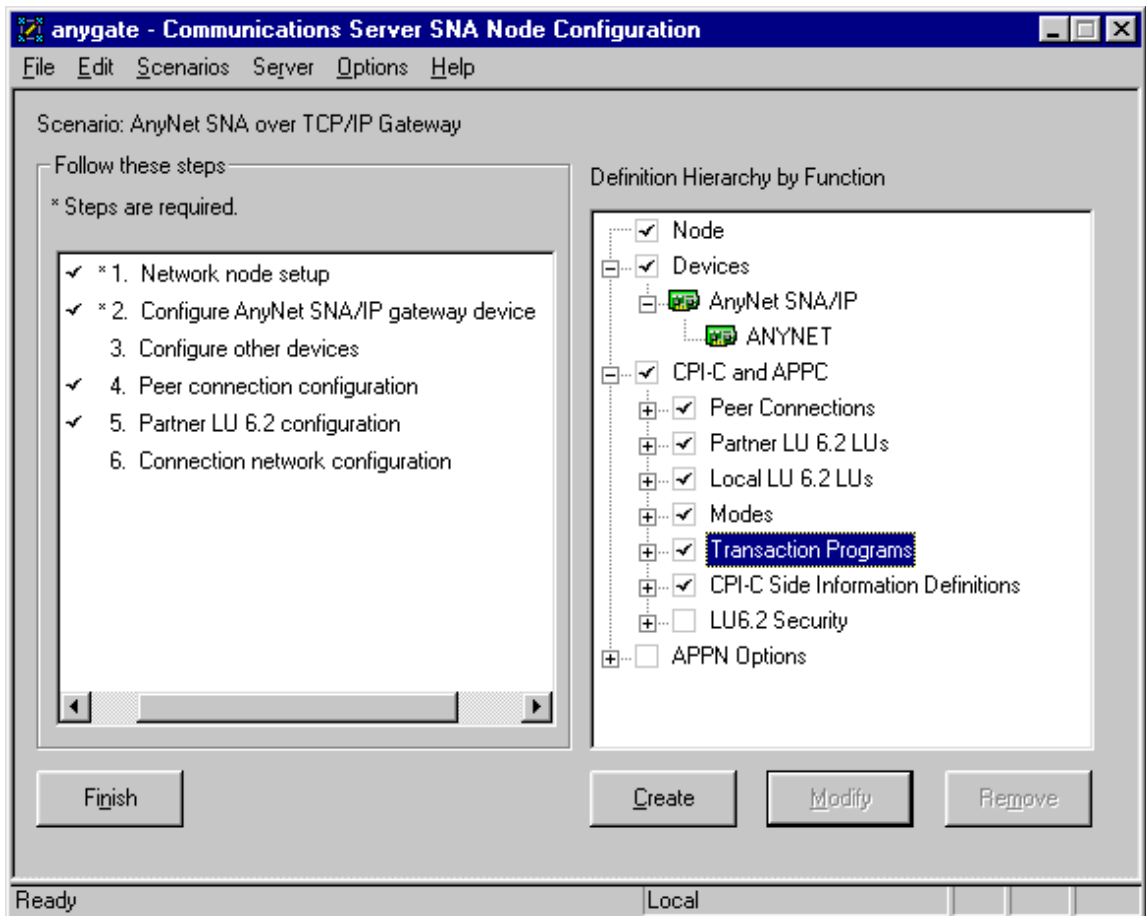
- Local LU name:** A text box containing "IBMDLC01".
- Dependent LU:** An unchecked checkbox.
- SNA API client use:** A checked checkbox.
- Local LU alias:** A text box containing "IBMDLC01".
- PU name:** A dropdown menu with a downward arrow.
- NAU address:** A dropdown menu with a downward arrow.

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

13. Accept the defaults for the **Advanced** tab (not shown), and then click the **OK** button to return to the Communications Server SNA Node.
14. Click within the **Modes** pane and then click the **Create** button. Define mode name IBMRDB as shown, and then accept the default values shown on the **Advanced** tab. Click the **OK** button to return to the Communications Server SNA Node Configuration window.



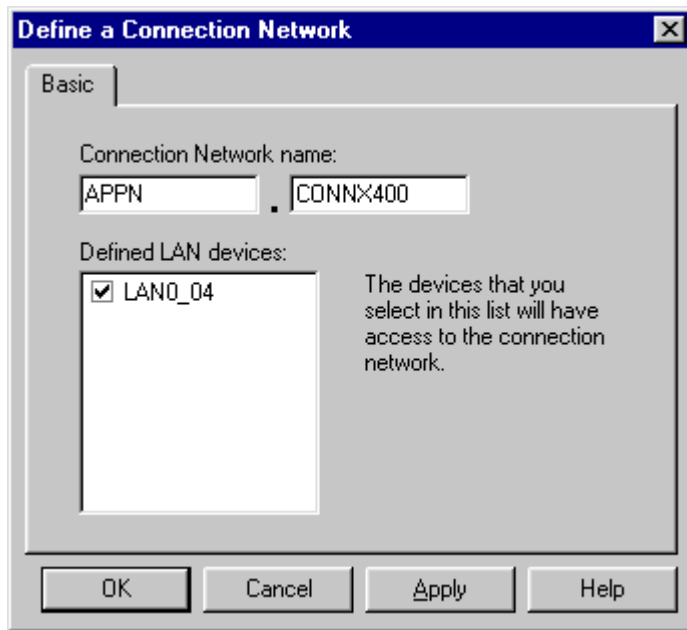
15. Select **Transaction Program** in the **CPI-C** and **APPC** node in the **Communications Server SNA Node Configuration** dialog box, and then click the **Create** button.



16. The **Define a Transaction Program** dialog box appears.

The screenshot shows the 'Define a Transaction Program' dialog box. The 'Basic' tab is active. The 'TP name' field contains '076DB'. The 'Service TP' checkbox is checked. The 'Complete pathname' and 'Program parameters' fields are empty. The 'Conversation type' dropdown is set to 'Basic', and the 'Synchronization level' dropdown is set to 'Any'. The 'Conversation security required' checkbox is unchecked. The 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom.

17. Enter the name of the transaction program in the **TP name** text box, and then check the **Service TP** check box. This is the name of the DRDA APPC transaction program, which is predefined at the target AS/400 system. Select the **Advanced** tab.
18. Verify that the **For SNA API Client Use** check box is checked.
19. Expand **APPN Options** and **APPN Connection Networks**. In the right pane, click the **Create** button.
20. The **Define a Connection Network** dialog box appears. Define the network name = netid.luname of the target OS/400. Click the **OK** button to return to the Communications Server SNA Node Configuration window.



21. Select **Save** on the **File** menu and start the connection.

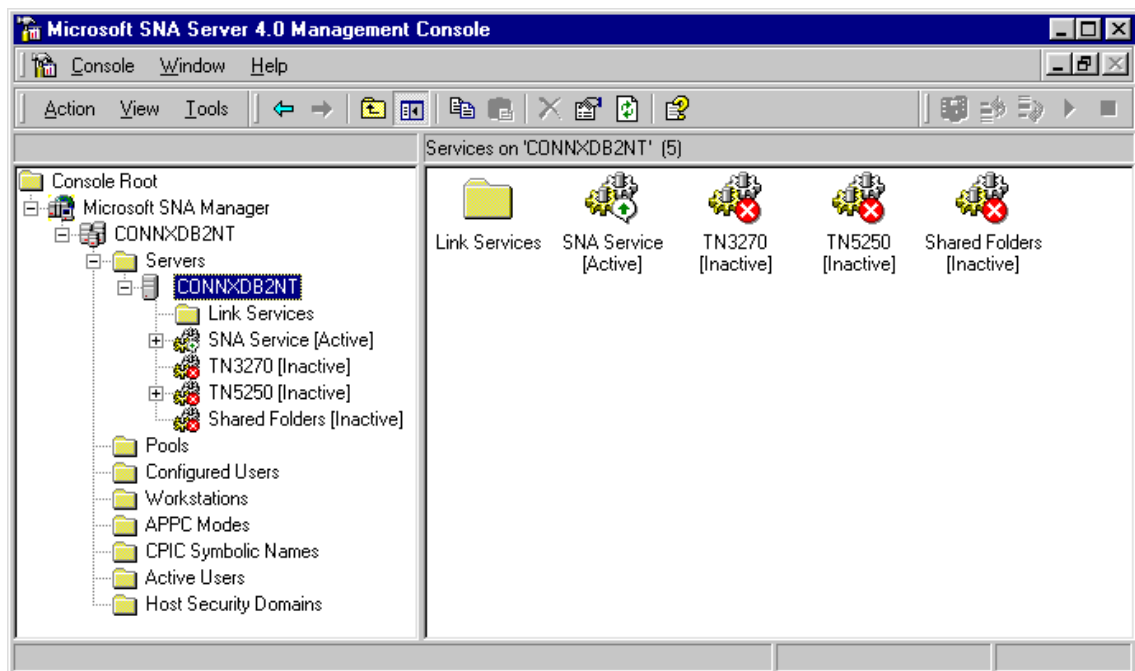
Communications Setup and Scenarios for MS SNA Server

To define the server node

The following presents an example of the CONNX MS SNA Server configuration, which connects to an OS/400 V4R2 machine via Ethernet 802.2 DLC (Data Link Control).

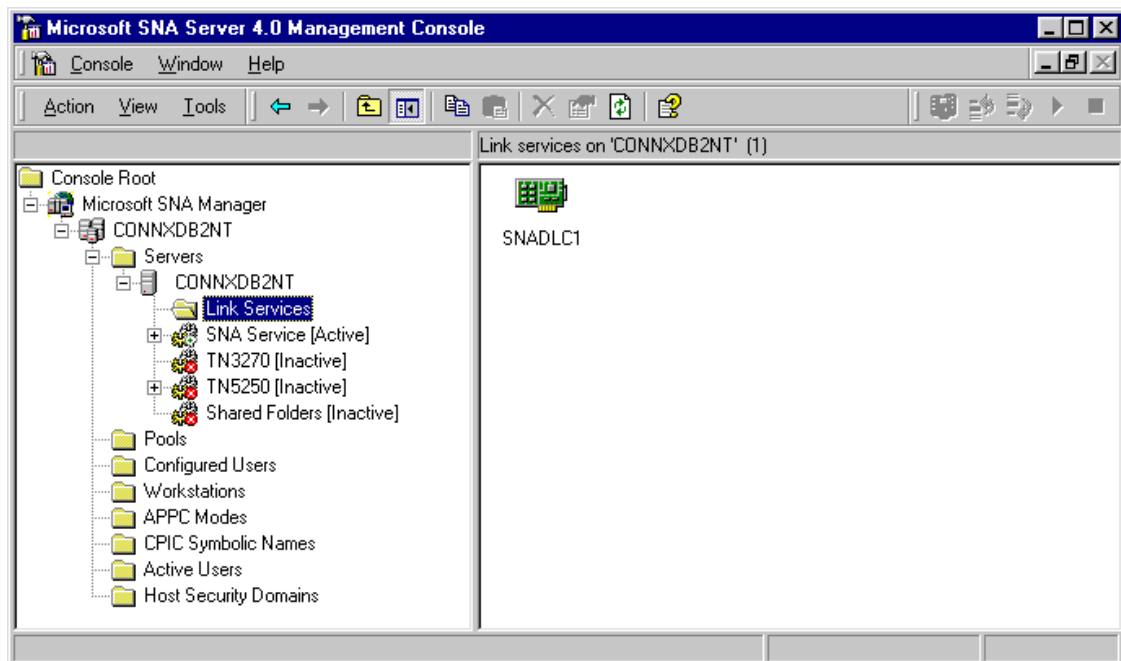
To define the server node for a Microsoft SNA Server 4.0

1. Start the **Microsoft SNA Server 4.0 Management Console** and define a server. In this example, the server is defined as CONNXDB2NT, and the fully qualified control point name is APPN.CONNXDB2.

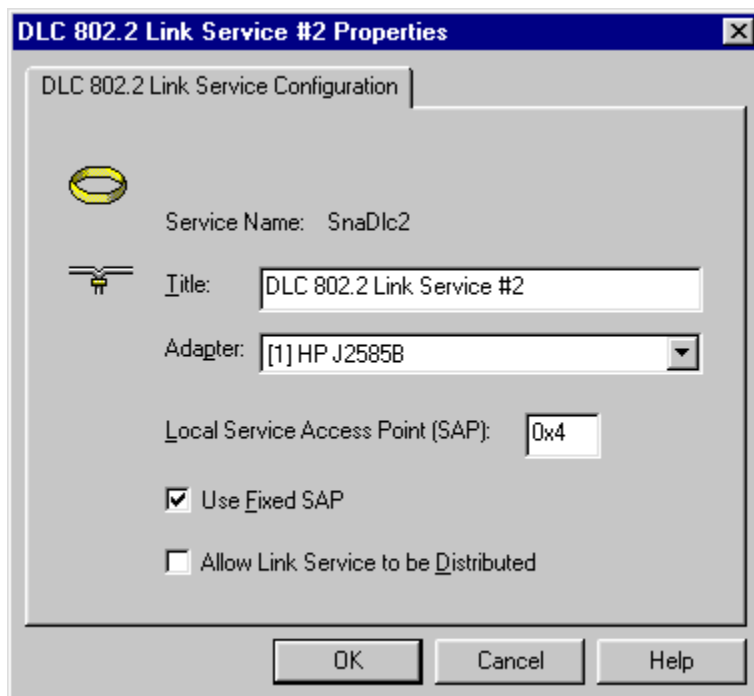


The screenshot shows a Windows-style dialog box titled "SNA Service: CONNXDB2NT Properties". It has a "General" tab selected. Inside the dialog, there is a server icon, a "Server Name" field containing "CONNXDB2NT", and an empty "Comment" text box. Below these is a section titled "SNA Network Control Point Name" which contains two sub-fields: "Network Name" with the value "APPN" and "Control Point Name" with the value "CONNXDB2". At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

2. Right click on the **Link Services Folder** in the left pane, select **New** and **Link Service** from the menu. Select **DLC 802.2 Link Service** in the **Insert Link Service** dialog box. Click the **Add** button.



3. The **DLC 802.2 Link Service Configuration** dialog box appears. Accept the defaults and click the **OK** button.



4. Right-click the **Connections** folder in the left pane. Select **New + 802.2** from the menu.
5. The **Connection Properties** dialog box appears. The displayed values are defaults, except that the Remote End has been changed from Host System to Peer System.

S102P6WM Properties

General | Address | System Identification | 802.2 DLC

Name: S102P6WM

Link Service: SnaDlc1 Ethernet

Comment:

Remote End:

- ☐ Host System
- ☒ Peer System
- ☐ Downstream
- ☐ PU Passthrough

Allowed Directions:

- ☒ Outgoing Calls
- ☐ Incoming Calls
- ☐ Both Directions

Activation:

- ☐ On Server Startup
- ☒ On Demand
- ☐ By Administrator

Passthrough via Connection: <None>

☐ Supports Dynamic Remote APPC LU Definition

OK Cancel Apply Help

6. Select the **Address** tab. The Remote Network Address text box contains an example value. The value of this field must match the LAN adapter address for the target system, which can be displayed via the OS/400 WRKLIND command. If you are not authorized to use this command, ask your OS/400 systems administrator for the target machine LAN adapter (MAC) address.

7. Select the **System Identification** tab. Note that the XID type is Format 3.

The screenshot shows the 'Properties' dialog box with the 'System Identification' tab selected. The '802.2 DLC' sub-tab is also visible. The 'Local Node Name' section contains the following fields:

- Network Name:** APPN
- Control Point Name:** CONNXDB2
- Local Node ID:** 05D FFFF

The 'Remote Node Name' section contains the following fields:

- Network Name:** (empty)
- Control Point Name:** (empty)
- Remote Node ID:** (empty)

The 'XID Type' section has two radio buttons: **Format 0** and **Format 3** (selected).

The 'Peer DLC Role' section has three radio buttons: **Primary**, **Secondary**, and **Negotiable** (selected).

The 'Compression Type' is set to **None** in a dropdown menu.

At the bottom of the dialog are four buttons: **OK**, **Cancel**, **Apply**, and **Help**.

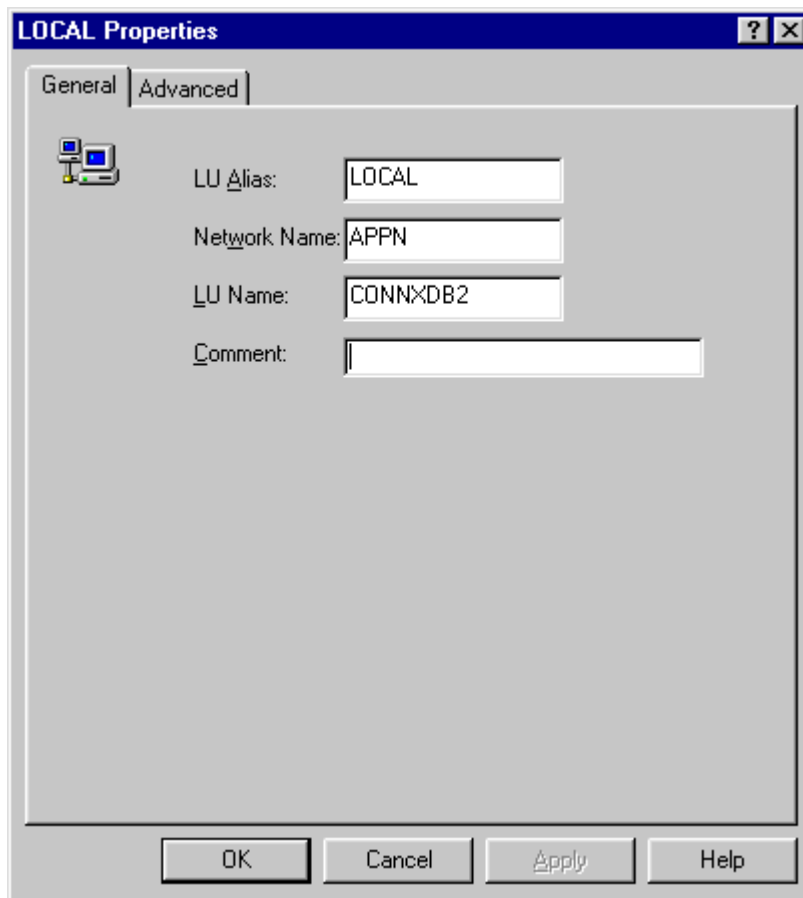
8. Select the **802.2 DLC** tab. The values displayed are the defaults.

The screenshot shows a Windows-style dialog box titled "Properties" with a blue header bar containing a question mark icon and a close button. Below the header is a tabbed interface with four tabs: "General", "Address", "System Identification", and "802.2 DLC". The "802.2 DLC" tab is selected. The main area of the dialog contains several configuration fields:

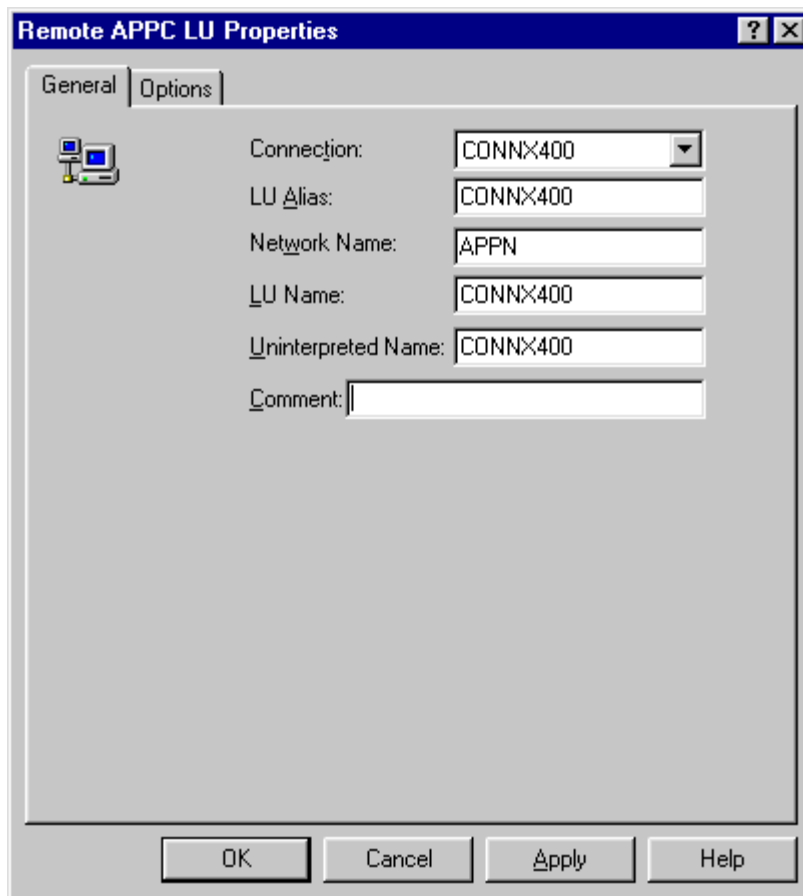
- "Max BTU Length:" with a text box containing "1493".
- "Receive ACK Threshold (frames):" with a spin box set to "6".
- "Unacknowledged Send Limit (frames):" with a spin box set to "8".
- "Retry Limit:" with a spin box set to "10".
- "XID Retries:" with a spin box set to "3".
- A group box titled "802.2 Timeouts" containing three dropdown menus:
 - "Response (t1):" set to "Default".
 - "Receive Ack (t2):" set to "Default".
 - "Inactivity (ti):" set to "Default".
- A group box titled "Connection Retry Limits" containing two dropdown menus:
 - "Maximum Retries:" set to "No Limit".
 - "Delay After Failure:" set to "Default".

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

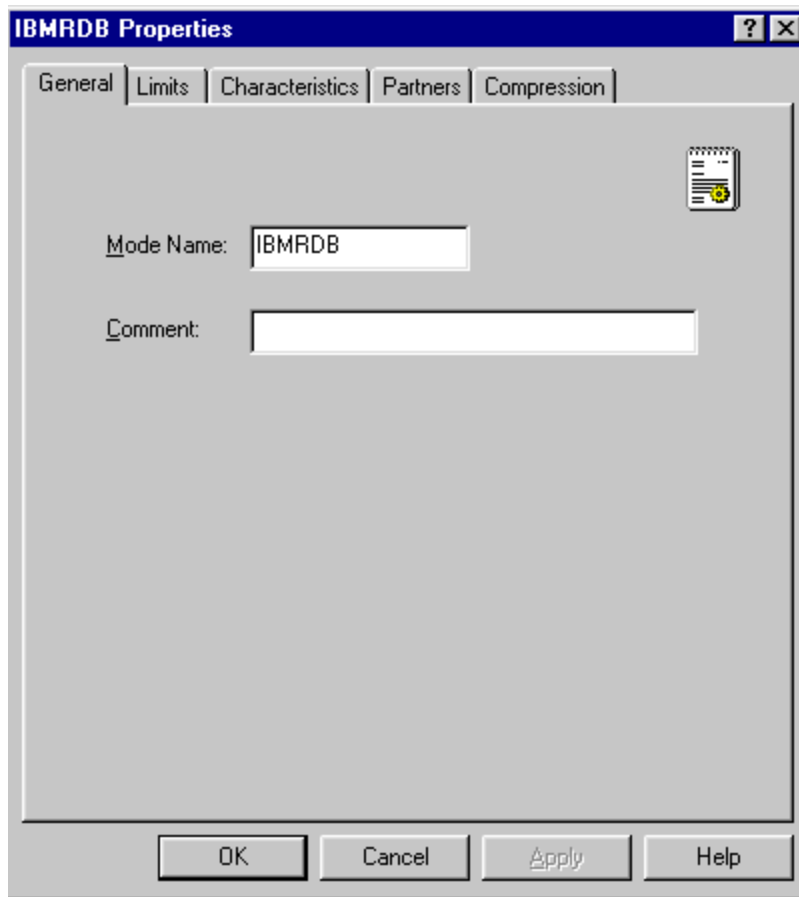
9. Right-click the **Local APPC LUs** folder, and then **Select New + Local LU** from the menu. The **Local Properties** dialog box appears.

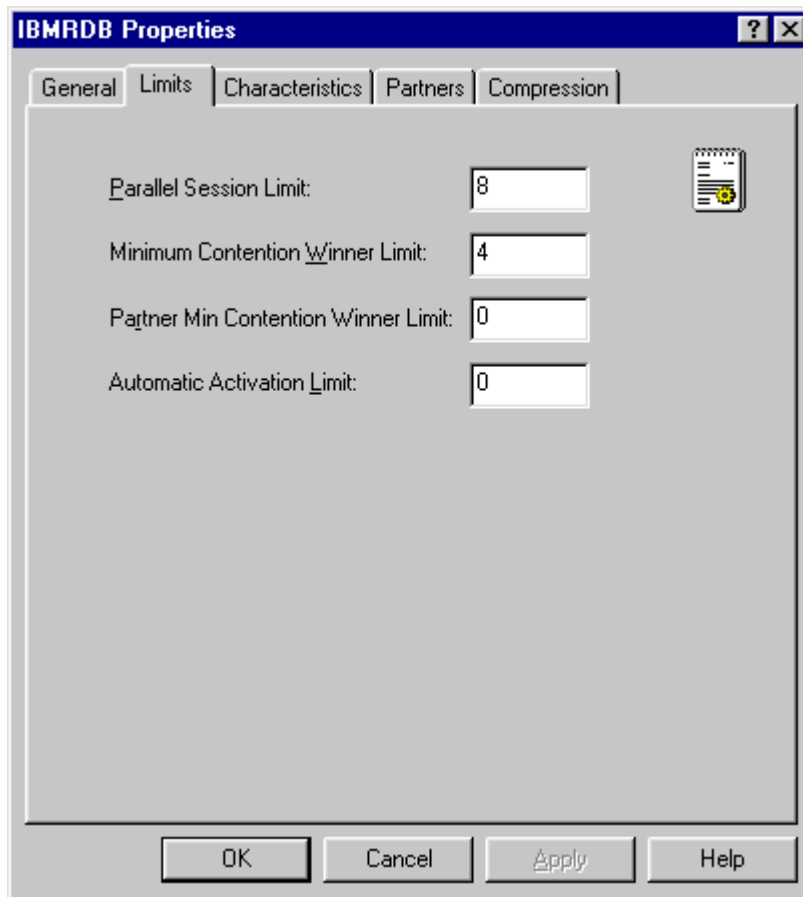


10. Select the **Advanced** tab (not shown); accept the default values.
11. Right-click the **Remote APPC LU** folder; then select **New + Remote LU** from the menu. The **Remote LU Properties** dialog box appears. Define 1- to 8-character names for **Network Name** and **LU Name**, which should match the Local Network ID and Local Control Point names as displayed by the OS/400 DSPNETA command. Define a **Partner LU** alias and an uninterpreted name, which in this example is CONNX400.



12. Right-click the **Remote APPC LU** folder; then select **New + Mode Definition** from the menu. The **APPC Mode Properties** dialog box appears. Enter appropriate values in the dialog boxes from the example screen shots. These values are compatible with the [IBMRDB](#) mode definition.





13. Your Microsoft SNA Server configuration is complete. You can test it by using the 5250 applet to connect to the host. If this connection works, you can connect to the OS/400 via this MS SNA server, provided that the CONNX DB2 module and the MS SNA Server client for Windows are installed on your client machine.

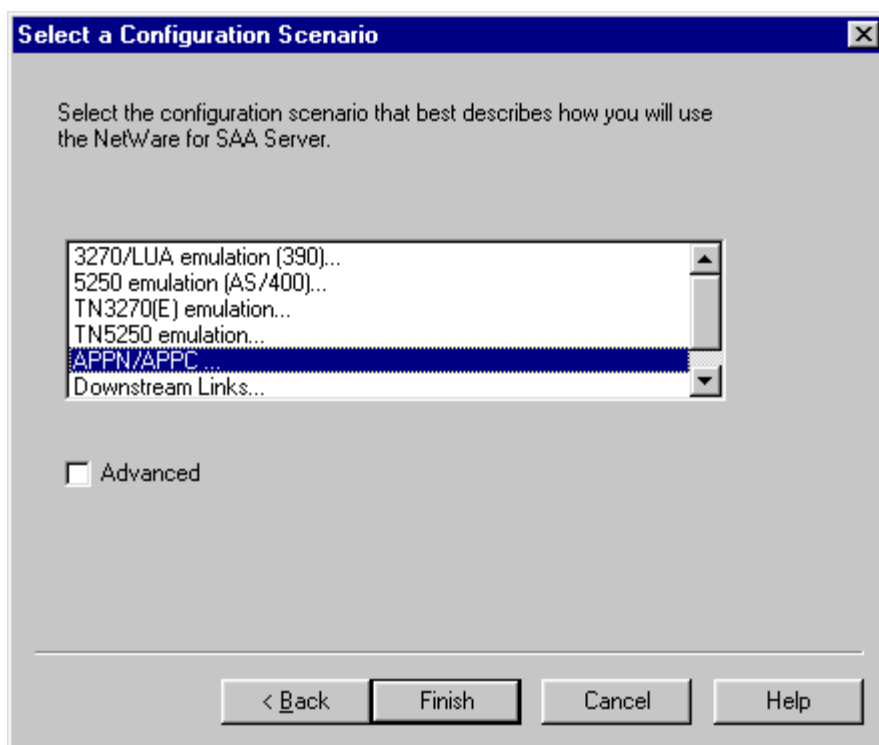
Communications Setup and Scenarios for Novell NetWare for SAA 4.0

To set up the Novell NetWare in an IEEE 802.2 DLC scenario

Note: To use this link via CONNX for DB2, you must have the NetWare for SAA 4.0 client software installed on the same client machine.

This example is based on a Novell NetWare for SAA 4.0 configuration installed on a Novell NetWare 5.0 server. The following scenario defines an IEEE 802.2 DLC (Data Link Control) connection to an AS/400 V4R2 machine.

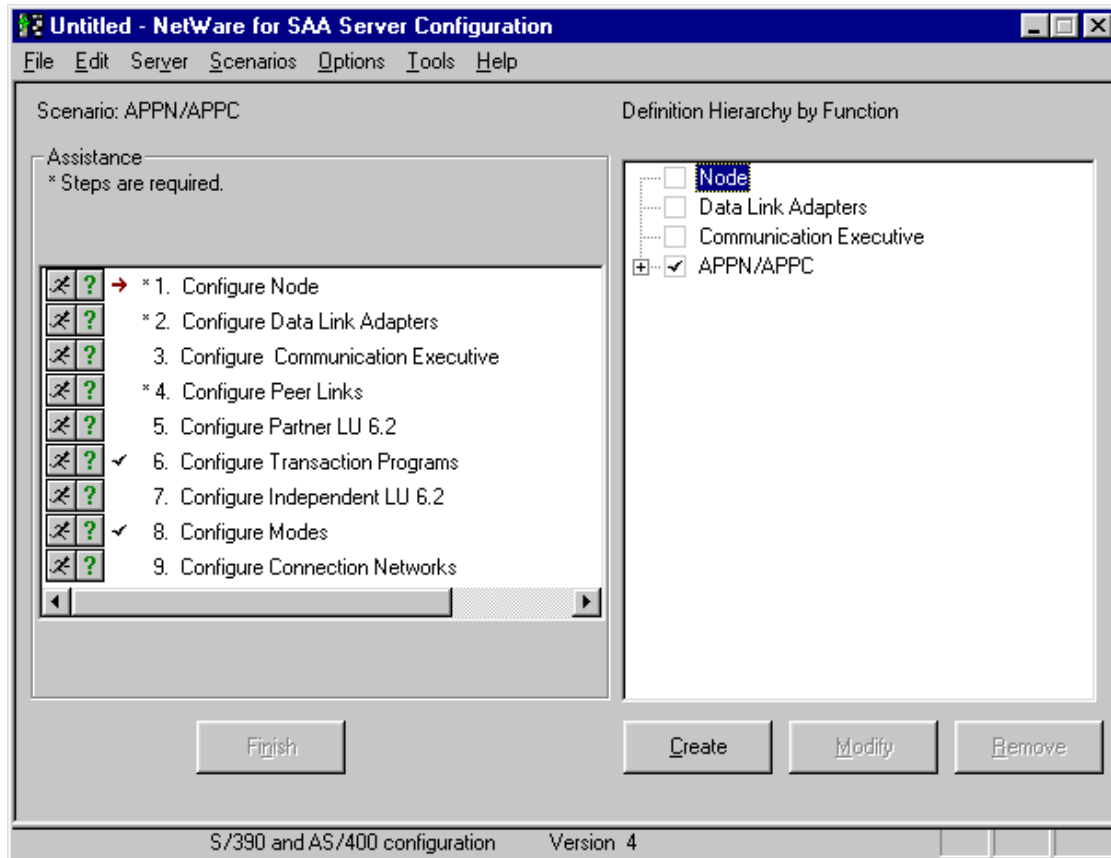
1. From a Microsoft Windows client machine on which the Novell NetWare client software is installed, start the NetWare for SAA Server configuration application by selecting **NetWare for SAA Management + NWSAA Server Configuration** from the **Start** menu.
2. Log in to the NetWare server. The **Select Configuration File** dialog box appears.
3. Select **New** or **Existing Configuration File**, and then click the **Next** button. The **Select a Configuration Scenario** dialog box appears.



4. Select **APPN/APPC** from the list box, and then click the **Finish** button.

To configure the node in APPN/APPC for Novell NetWare

1. Click **Node** in the **Definition Hierarchy by Function** pane, and then click the **Create** button.



2. The **Configure the Node** dialog box appears.

Configure the Node

Basic | Advanced | DLU Requester | Encryption Keys

NetWare for SAA version: 4.0.0

Control Point (CP)

Network name: APPN . CP name: SAA40ETH

CP alias: SAA40ETH

Local Node ID

Block ID: 05D Physical Unit ID: 00000

Node Type

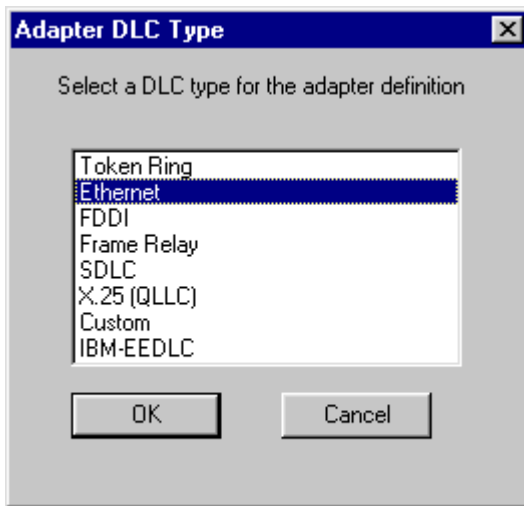
☒ End Node
☐ Network Node
☐ Branch Extender Node

OK Cancel Apply Help

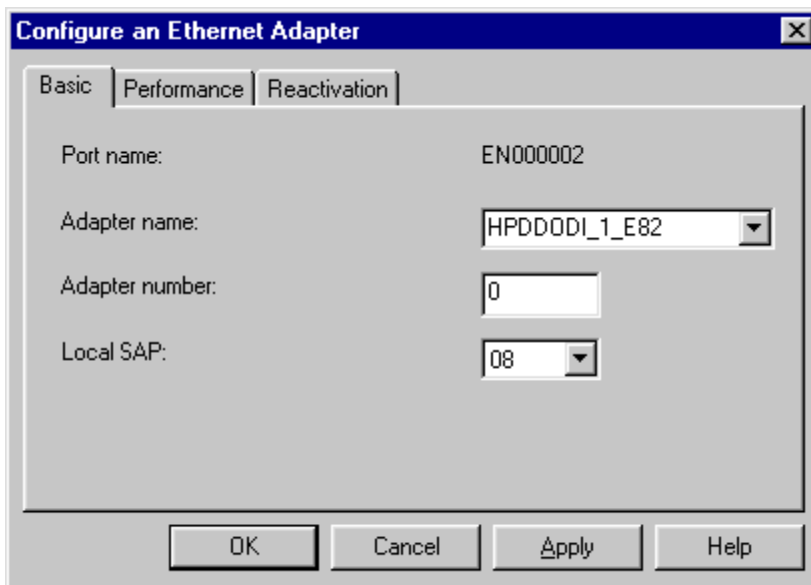
3. The values for network name, CP (Control Point) name, and CP alias are user-defined. Note that the **Local Node Block ID** has been changed from 017 to 05D.
4. Accept the defaults for the **Advanced**, **DLU Requester**, and **Encryption Keys** dialog box tabs.

To configure Data Link Adapters

1. From the **Definition Hierarchy by Function** pane, select **Data Link Adapters**, and then click the **Create** button. The **Adapter DLC Type** dialog box appears.



2. Select **Ethernet**, and then click the **OK** button.



3. For this example, an adapter name has been defined at the NetWare for SAA server. Accept the defaults for **Adapter Number** and **Local SAP**, and for the **Performance** and **Reactivation** dialog box tabs.

To define a peer link

1. From the **Definition Hierarchy by Function** pane, expand the **APPN/APPN** node, and then select the **Peer Links** node. Click the **Create** button. The **Link Type** dialog box appears.
2. For this example, accept the **Defined** data link adapter type = Ethernet, and click the **OK** button. The **Configure an Ethernet Peer Link** dialog box appears.

Configure an Ethernet Peer Link

Basic | Adjacent Node Parameters | Advanced | Performance | Reactivation

Peer link name: LINK0000

Port name (adapter): EN000001 (HPDDODI_1_E82) 04

Discover network addresses...

Remote node address: 020000000001

Remote SAP: 04

☐ Reverse address bytes

OK Cancel Apply Help

3. For this example, the **Remote Node Address** text box on the **Basic** tab matches the local adapter address of the target system. ([Verifying IEEE 802.2 DLC configuration](#))
4. Select the **Advanced** tab. Enable **APPN** support, and then enter a block ID = 05D and a node ID in the range of 00000 to FFFFF.

The screenshot shows the 'Configure an Ethernet Peer Link' dialog box with the 'Advanced' tab selected. The dialog has five tabs: 'Basic', 'Adjacent Node Parameters', 'Advanced', 'Performance', and 'Reactivation'. The 'Advanced' tab contains the following settings:

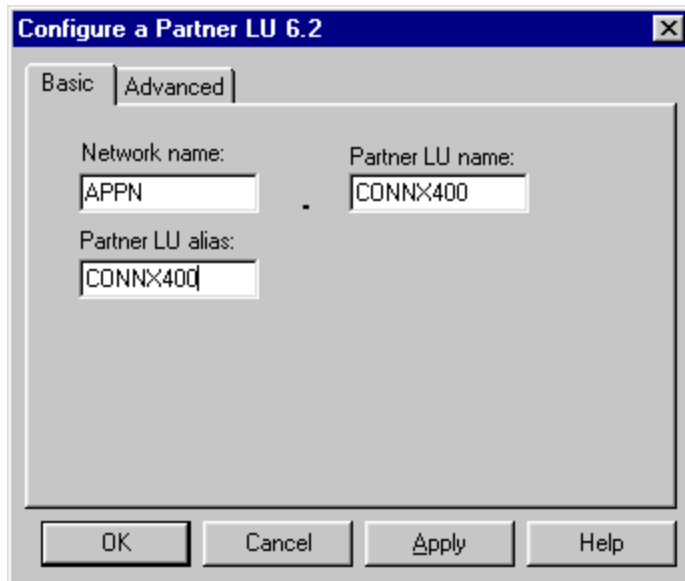
- ☒ Activate link at start
- ☐ Auto-activate support
- ☒ Link to preferred NN server
- ☐ HPR support
- ☒ APPN support
- Limited resource: No (dropdown menu)
- Link deactivation timer: 10 (text field)
- Local Node ID section:
 - Block ID: 05D (text field)
 - Physical Unit ID: 00000 (text field)

At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

5. Accept the defaults for the **Adjacent Node Parameters**, **Performance**, and **Reactivation** tabs (not shown).

To define a Partner LU 6.2

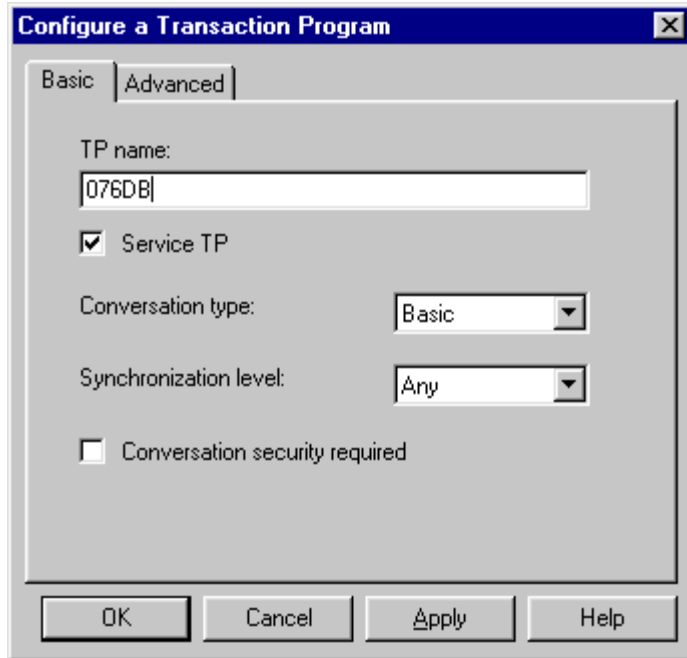
1. From the **Definition Hierarchy by Function** pane, select the **Partner LU 6.2** node, and then click the **Create** button. The **Configure a Partner LU 6.2** dialog box appears.
2. Enter the **NetID** and the system name of the target OS/400 database in the **Network name** and **Partner LU name** text boxes. This information can be displayed via the DSPNETA command.
3. Define a **Partner LU Alias**, which in this example is the same as the Partner LU name = CONNX400. (Example only.) Check with your system administrator for the netid.luname of the target OS/400.



4. Accept the defaults on the **Advanced** tab.

To define a Transaction Program Name

1. From the **Definition Hierarchy by Function** pane, click the **Transaction Programs** node, and then click the **Create** button. The **Configure a Transaction Program** dialog box appears.
2. Enter the name of the Transaction program = 076DB in the TP name text box, and then check the **Service TP** check box. This is the name of the DRDA APPC transaction program, which is predefined at the target AS/400 system.
3. Accept the defaults on the **Advanced** tab.



The image shows a Windows-style dialog box titled "Configure a Transaction Program". It has two tabs: "Basic" and "Advanced", with "Advanced" currently selected. Inside the dialog, there is a text box labeled "TP name:" containing the text "076DB". Below this is a checked checkbox labeled "Service TP". Further down are two dropdown menus: "Conversation type:" set to "Basic" and "Synchronization level:" set to "Any". At the bottom of the dialog area is an unchecked checkbox labeled "Conversation security required". At the very bottom of the dialog box are four buttons: "OK", "Cancel", "Apply", and "Help".

To define a mode

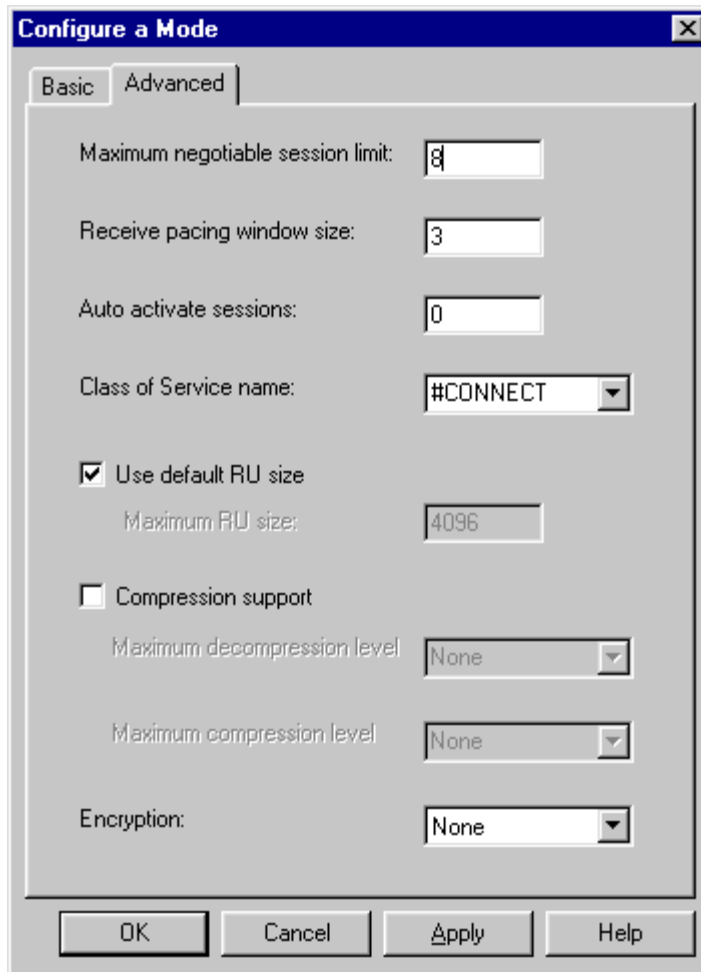
1. From the **Definition Hierarchy by Function** pane, click the **Modes** node, and then click the **Create** button. The **Configure a Mode** dialog box appears.

For DRDA connections, the target AS/400 system has a predefined mode name = IBMRDB. Define this mode name, and then enter the mode session limits and minimum contention winner sessions as shown. Note that these values match the values for maximum sessions and (maximum sessions - locally controlled sessions) as shown in the output of the DSPMODD command for [IBMRDB](#).



The screenshot shows the 'Configure a Mode' dialog box with the 'Basic' tab selected. The dialog has a title bar with a close button. Inside, there are two tabs: 'Basic' and 'Advanced'. The 'Basic' tab contains three input fields: 'Mode name:' with the value 'IBMRDB', 'Mode session limit:' with the value '8', and 'Minimum contention winner sessions:' with the value '4'. At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

2. Select the **Advanced** tab, and then enter the values for maximum negotiable session limit and Receive pacing window size. Note that these values match the values for maximum sessions and Outbound pacing value, as shown in the output of the DSPMODD command for [IBMRDB](#).



The image shows a 'Configure a Mode' dialog box with two tabs: 'Basic' and 'Advanced'. The 'Advanced' tab is selected. The dialog contains several configuration options:

- Maximum negotiable session limit: 8
- Receive pacing window size: 3
- Auto activate sessions: 0
- Class of Service name: #CONNECT
- ☒ Use default RU size
 - Maximum RU size: 4096
- ☐ Compression support
 - Maximum decompression level: None
 - Maximum compression level: None
- Encryption: None

At the bottom of the dialog are four buttons: OK, Cancel, Apply, and Help.

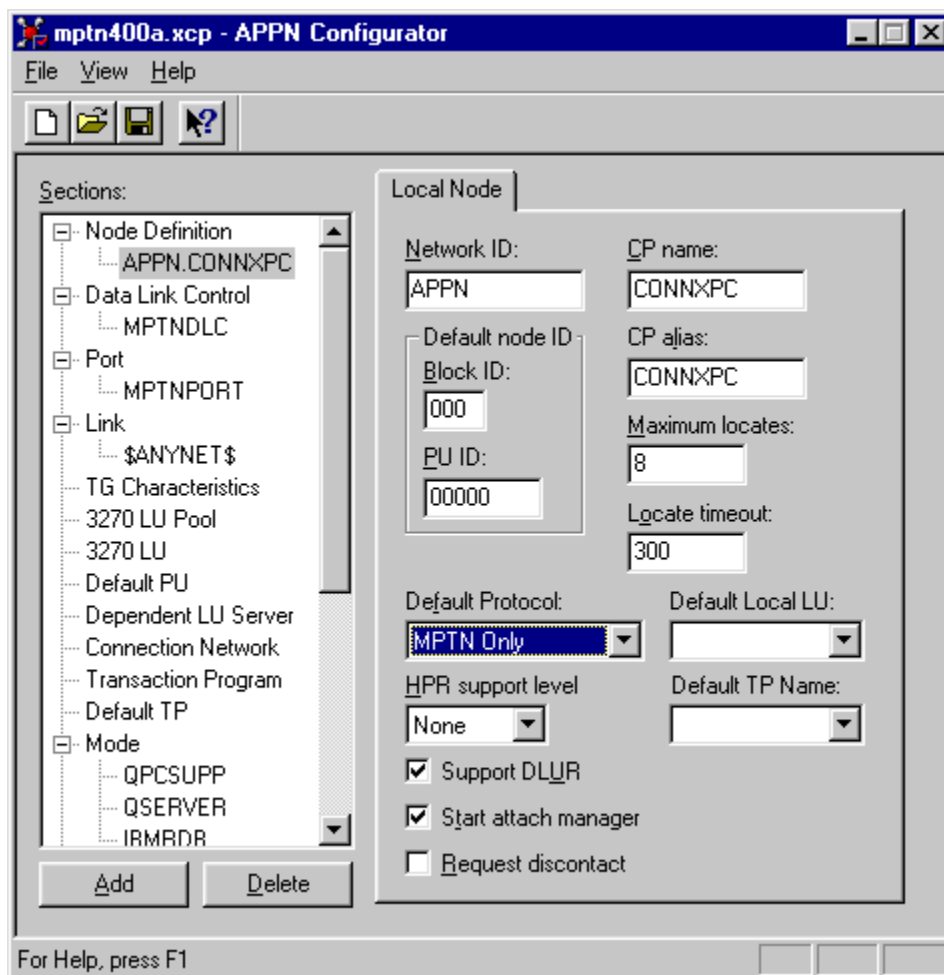
3. Your NetWare for SAA 4.0 configuration for IEEE 802.2 DLC access via Ethernet to an AS/400 is complete.

Communications Setup and Scenarios for 2-Tier SNA Client Configuration

To set up 2-tier SNA client configuration for Attachmate - MPTN

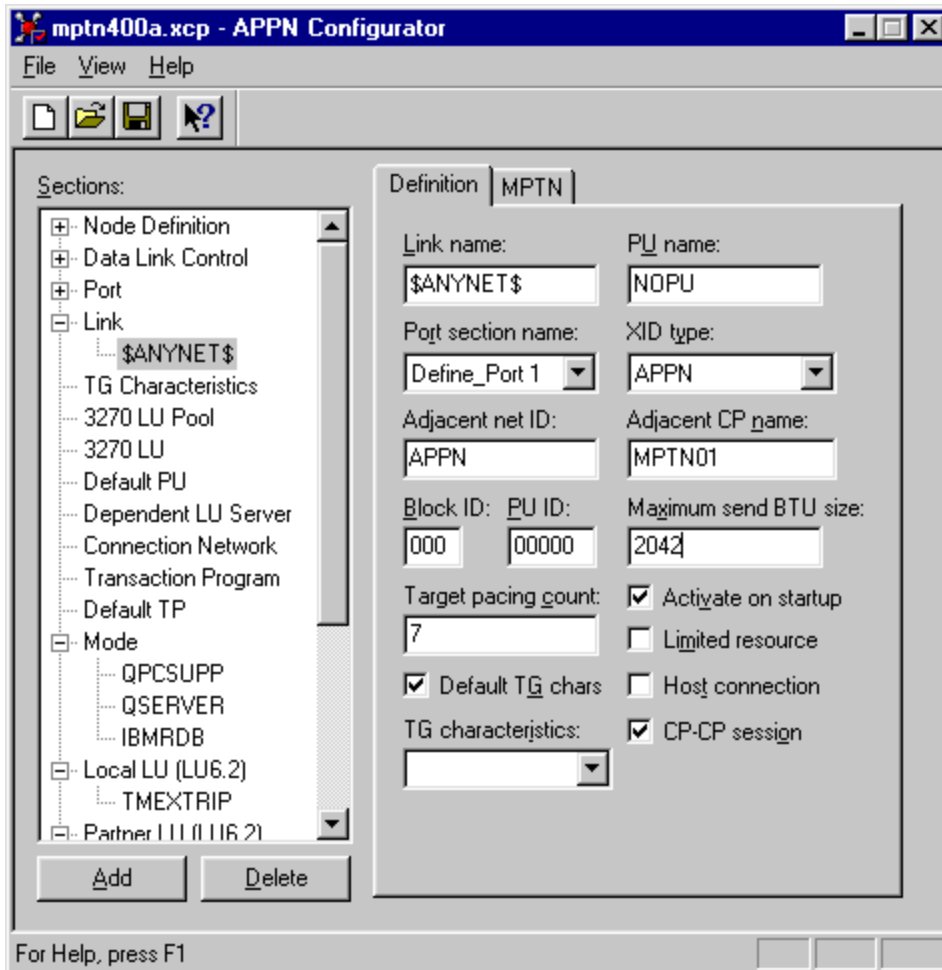
Several SNA vendors implement two-tier direct-connect products, including Attachmate and RUMBA. You can configure CONNX to work with these SNA clients, provided that the CONNX client is installed on the same client PC as the SNA vendor client. CONNX detects the supported SNA client stacks automatically.

1. Select **Start**, point to **Programs**, and then select the Attachmate Extra! Personal Client 32-bit or Extra! Office 32-bit Client application. Select **APPN Node+APPN Configurator** from the menu.
2. Under **Sections**, select **Node Definition**, and then click the **Add** button. The **Local Node** tab appears. The values in this tab define the local computer to the network. For this example, the network name is APPN; the CP (Control Point) Name is CONNXPC. Select **Data Link Control** under **Sections**, and then click the **Add** button.

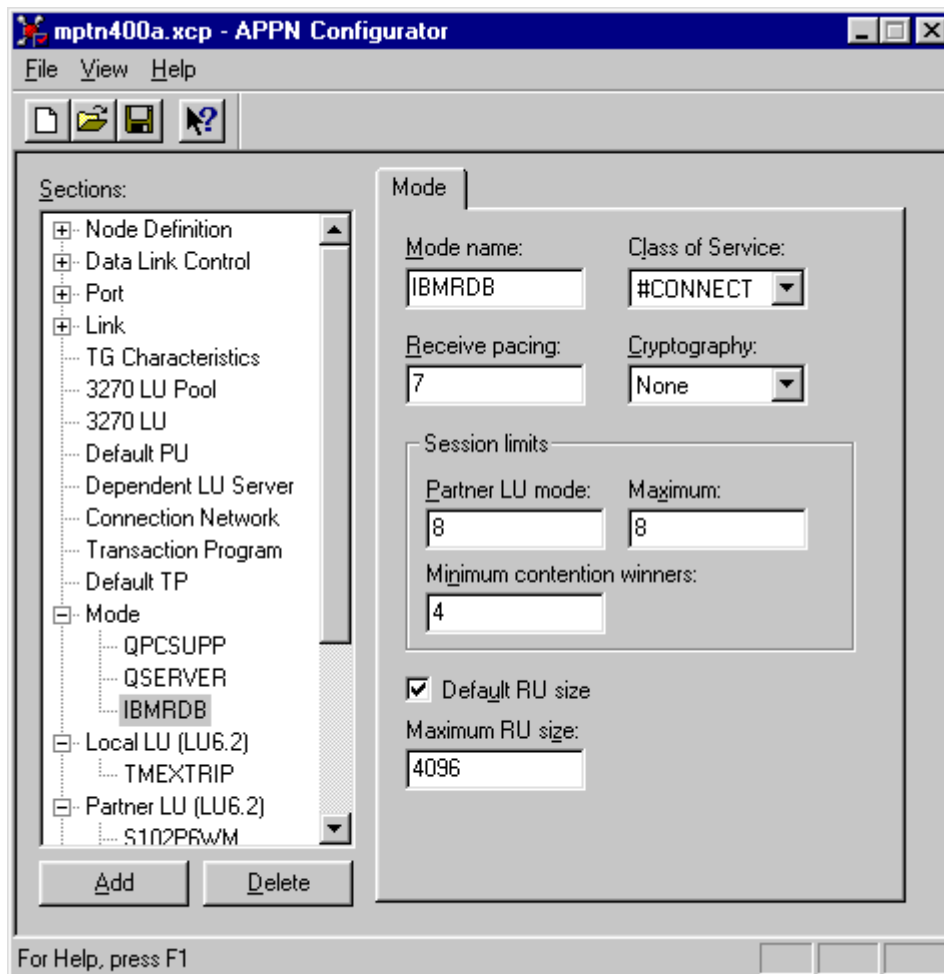


3. The **Add Data Link Control** dialog box appears. Select **MPTN** as the DLC type in the **DLC Type** list box, and then click the **OK** button. The MPTN tab appears in the APPN Configurator window. Enter a symbolic name, in this case MPTNDLC.
4. Select **Port** under **Sections** in the APPN Configurator window, and then click the **Add** button. The **Add Port Definition** dialog box appears. Select **MPTN** as the DLC Type in the **DLC Type** list box, and then click the **OK** button to return to the APPN Configurator window. The **Port** tab appears in the right pane. Representative information is complete.

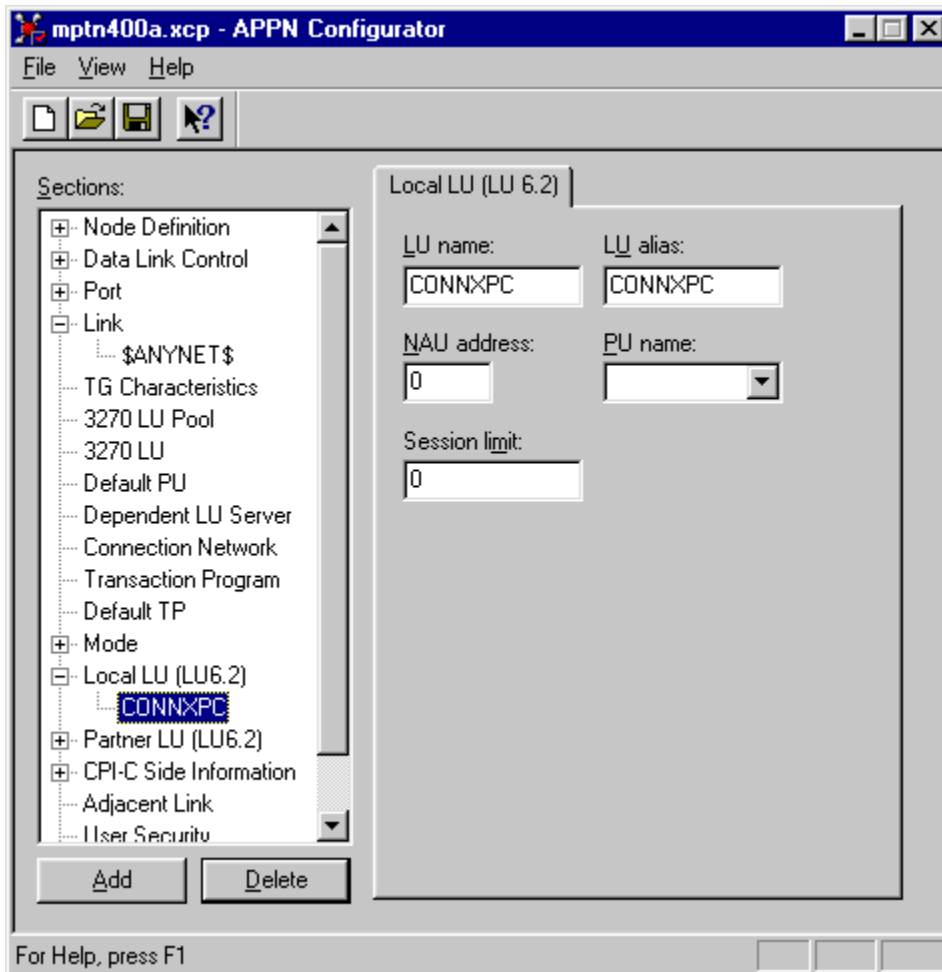
5. Select **Link** under **Sections**, and then click the **Add** button. The **Add Link Definition** dialog box appears. Select **MPTN** as the DLC link type in the **DLC Type** list box, and then click the **OK** button to return to the APPN Configurator window.
6. The **Definition Tab** appears in the right pane. Accept the defaults.



7. Select the **MPTN** tab in the right pane. Check the **Swap NetID/LU Name** in the **Domain** check box. (not shown)
8. Select **Mode** under **Sections**, and click the **Add** button. The **Mode** tab appears. Enter the **IBMRDB mode name** in the **Mode Name** text box. These values are compatible with the IBMRDB mode definition as displayed by the OS/400 DSPMODD command.



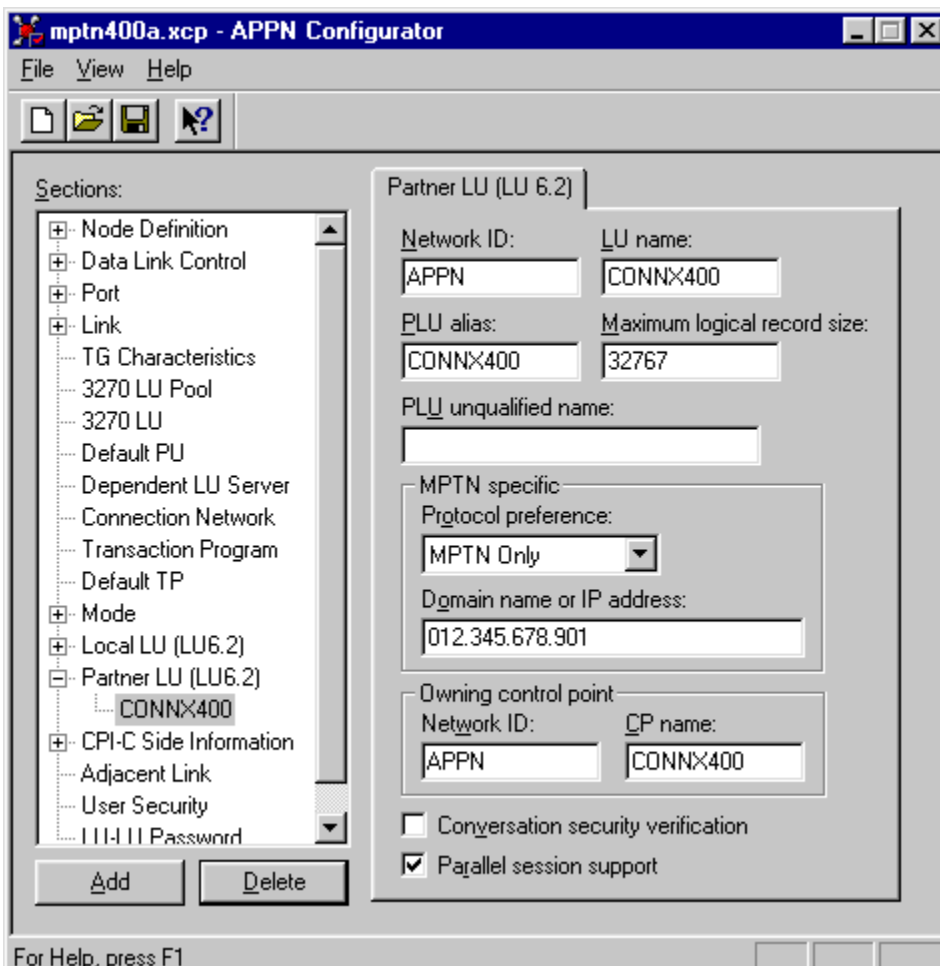
9. Select **Local LU 6.2** under **Sections**, and then click the **Add** button. The Local LU (LU 6.2) tab appears. In keeping with previous examples, both the Local LU name and the Local LU alias are defined as CONNXPC. Note that the NAU address field is 0; this value defines an independent local LU.



10. Select **Partner LU (LU 6.2)** under **Sections**, and then click the **Add** button. The **Partner LU (LU 6.2)** tab appears. As in previous examples, the target AS/400 is defined as APPN.CONNX400, as is the owning control point name. The Attachmate MPTN driver maps the dotted TCP/IP address to a netid.luname via the hosts file, which is located in either the \winnt\system32\drivers\etc or the \windows directory for Windows. For example:

012.345.678.901 connx400.appn.sna.ibm.com #Anynet

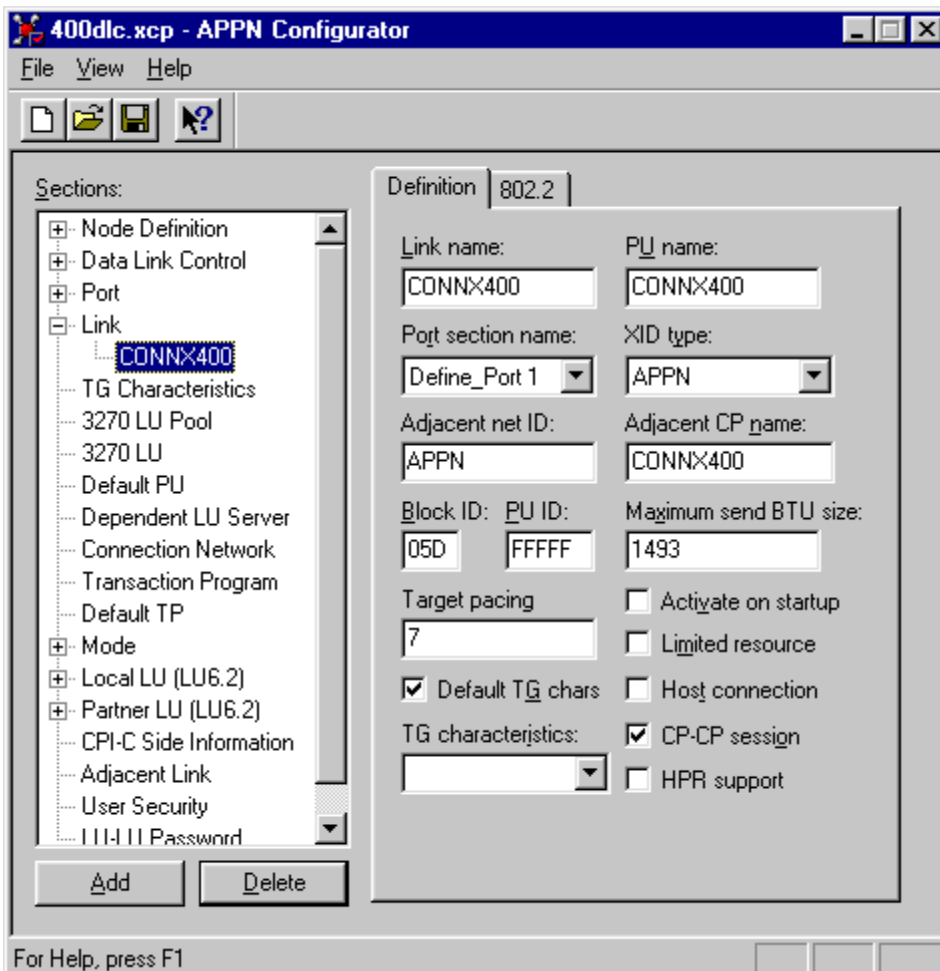
Note that the host file transposes the netid.luname. The sna.ibm.com suffix is discarded.



11. The **Attachmate MPTN** configuration is complete. These values are used later in the CONNX client configuration dialog box.

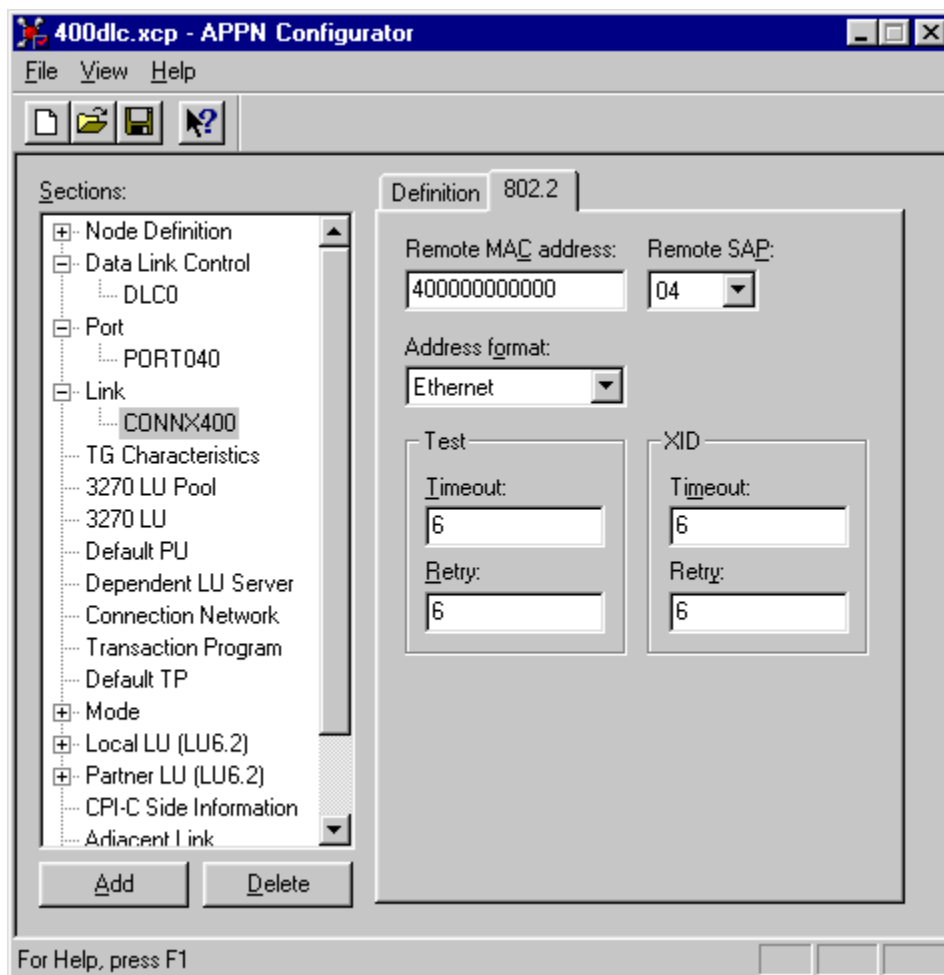
To set up 2-tier client configuration for Attachmate - IEEE 802.2

1. Select **Start**, point to **Programs**, and then select the **Attachmate Extra! Personal Client 32-bit** or **Extra! Office 32-bit** client application. Select APPN Node+APPN Configurator from the menu.
2. Under **Sections**, select **Node Definition**, and then click the **Add** button. The **Local Node** tab appears. The values in this tab define the local computer to the network. Select **Data Link Control** under **Sections**, and then click the **Add** button. Select **Network (802.2)** as the DLC type in the list box. Click the **OK** button to return to the APPN Configurator window.
3. The **802.2** tab appears. Choose a user-defined DLC name, in this case DLCO. Accept the defaults.
4. Under **Sections**, select **Port**, and then click the **Add** button. The **Add Port Definition** dialog box appears. Select **Network (802.2)** as the DLC type in the list box, and then click the **OK** button to return to the APPN Configurator window.
5. Under **Sections**, select **Link**, and then click the **Add** button. The **Add Link Definition** dialog box appears. Select **Network (802.2)** as the link type in the list box, and then click the **OK** button to return to the APPN Configurator window.
6. The **Definition** tab appears with representative values. For purposes of this example, the target host is defined as APPN.CONNX400 in the adjacent NetID and CP Name text boxes. You can substitute the fully qualified netid.luname of the target AS/400 at your installation.

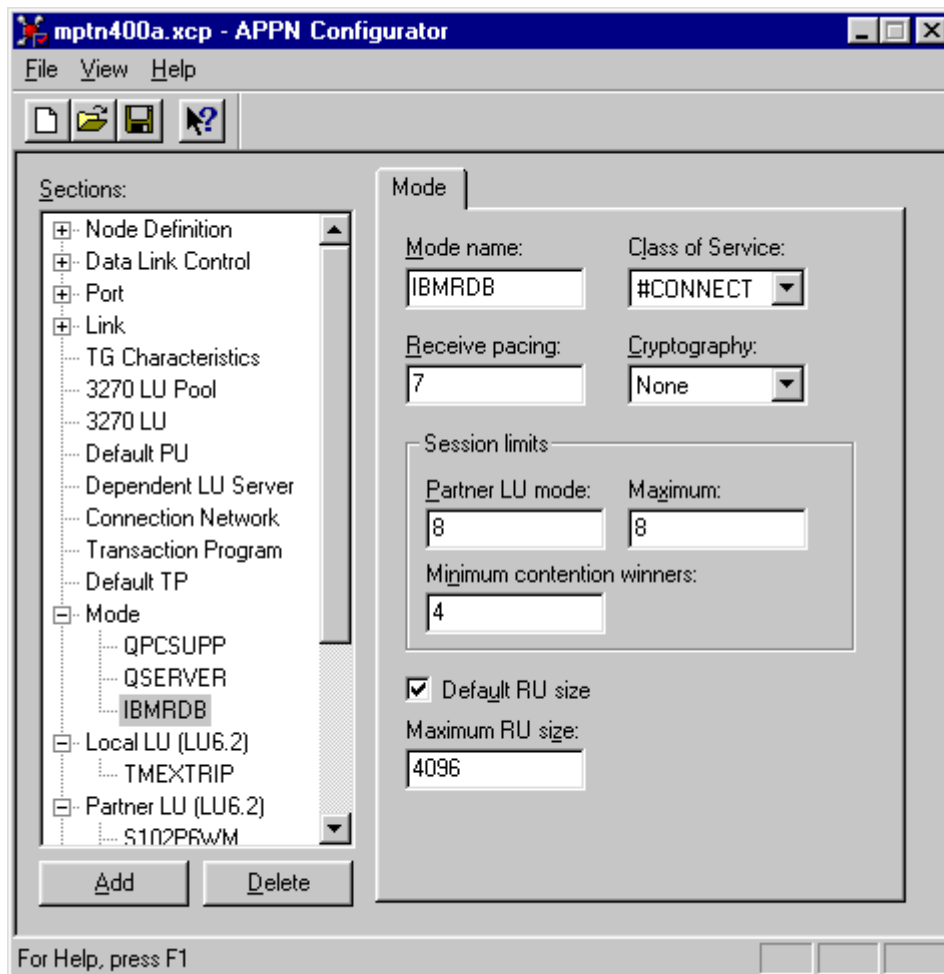


7. Select the **802.2** tab in the APPN Configurator window. Enter the destination address of the host machine. The value of this field must match the LAN adapter address for the targetsystem, which

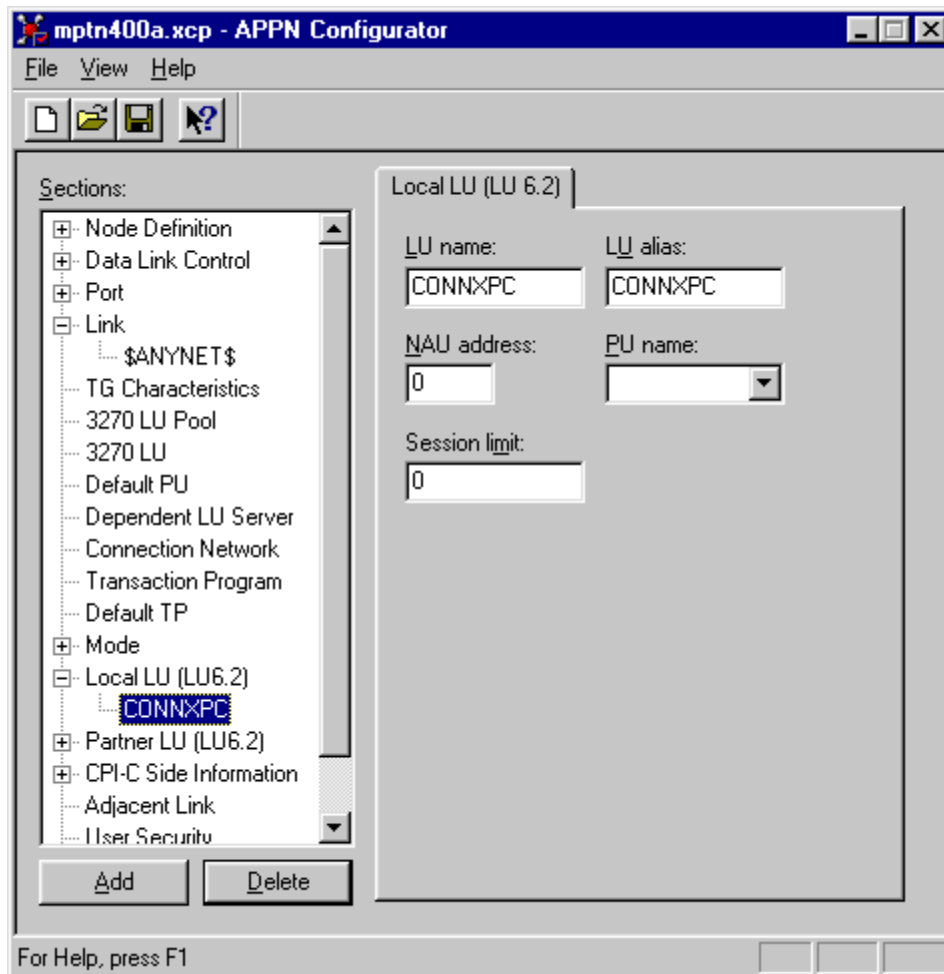
can be displayed via the OS/400 WRKLIND command. Refer to the screen shot in [To verify IBMRDB mode description](#) for an example. If you are not authorized to use this command, ask your OS/400 systems administrator for the target machine LAN adapter (MAC) address.



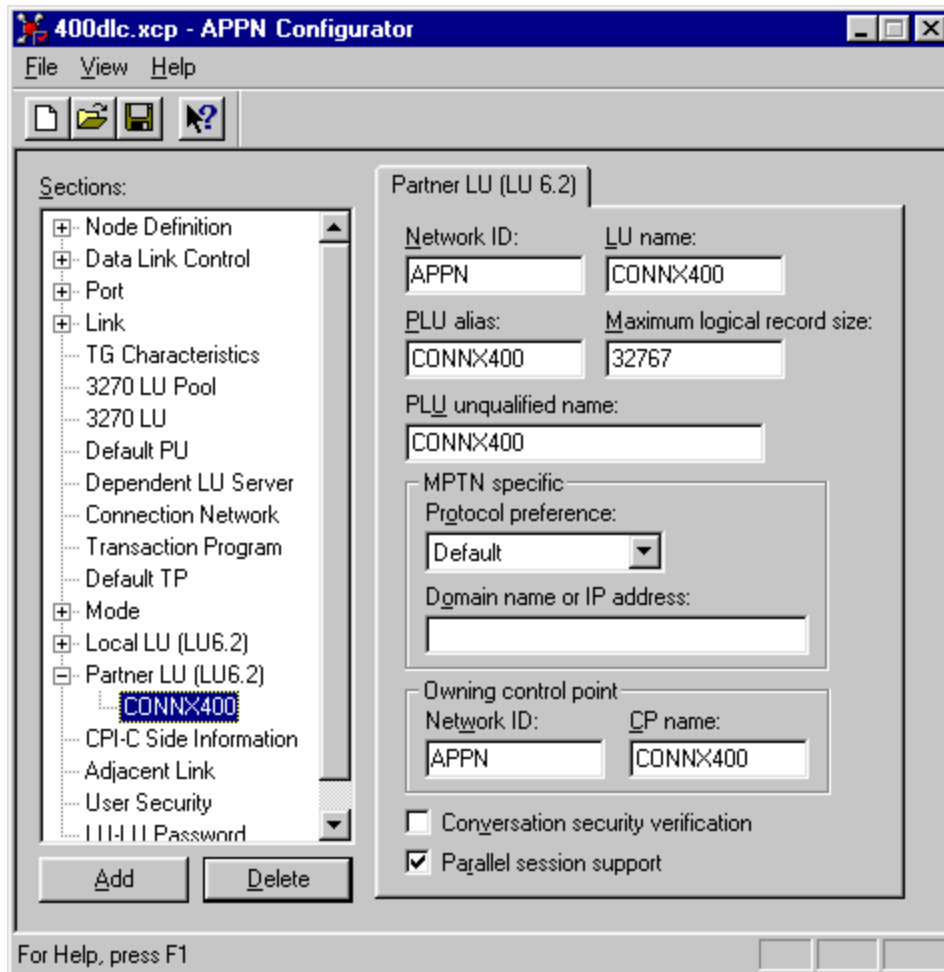
8. Under **Sections**, select **Mode**, and then click the **Add** button. The **Mode** tab appears. Enter the **IBMRDB mode name** in the **Mode Name** text box. Representative values are entered. These values are compatible with the IBMRDB mode definitions as displayed by the OS/400 DSPMODD command.



9. Under Sections, select **Local LU 6.2 (LU 6.2)**, and then click the **Add** button. The **Local LU (LU 6.2)** tab appears. In keeping with previous examples, both the Local LU name and the Local LU alias are defined as CONNXPC. Note that the NAU address field is 0; this value defines an independent local LU.



10. Under **Sections**, select **Partner LU (LU 6.2)**, and then click the **Add** button. The **Partner LU (LU 6.2)** tab appears. As in previous examples, the target AS/400 is defined as APPN.CONNX400, as is the owning control point name.



11. Your Attachmate IEEE 802.2 DLC configuration is complete. These values are used later in the CONNX client configuration dialog box.

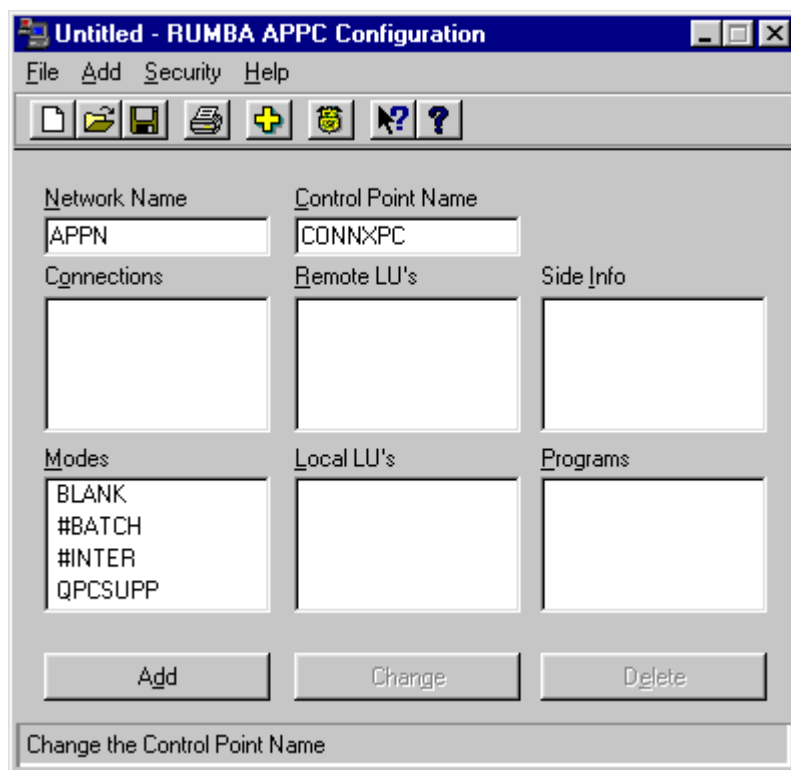
To set up 2- tier SNA client configuration for RUMBA MPTN

This section assumes you have the RUMBA product of Netmanage, Inc., installed on your computer.

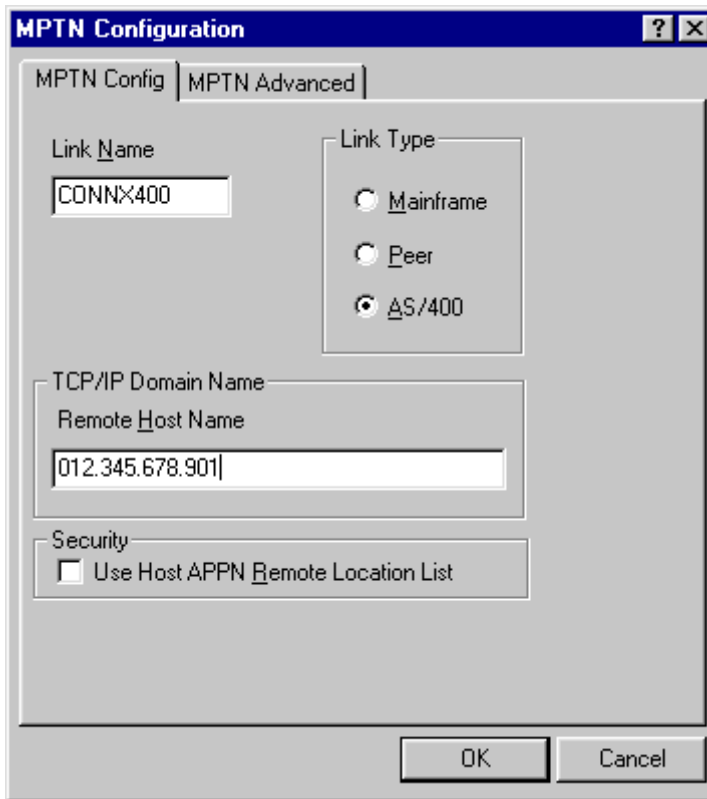
1. Select **Start**, point to **Programs**, and then select the **RUMBA APPC** configuration utility. The RUMBA APPC Configuration window appears.

For this example, enter **APPN** as the network name.

2. Type the user-defined 1- to 8-character name of your client computer in the **Control Point Name** text box.

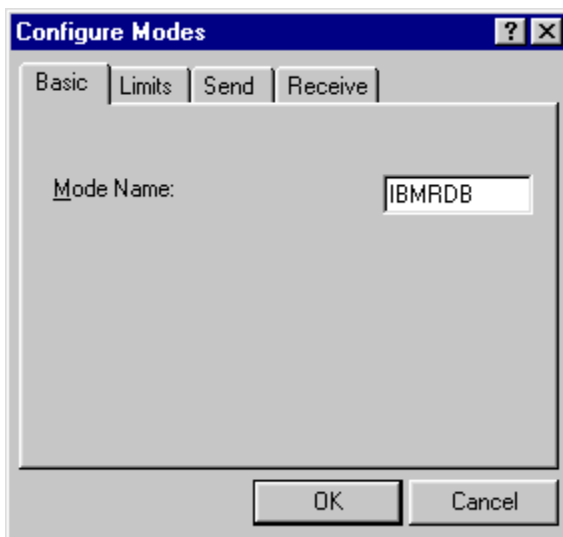


3. Click within the **Connections** pane, and then click the **Add** button. The **Select a Connection** dialog box appears. Select **MPTN**, and then click the **OK** button.
4. The **MPTN Configuration** dialog box appears. Select the **MPTN Config** tab. Under **Link Name**, type a user-defined 1- to 8-character connection name. Under **TCP/IP Domain Name**, type the dotted TCP/IP address of the host machine or the symbolic name. Verify that the **AS/400** option is selected under **Link Type**.



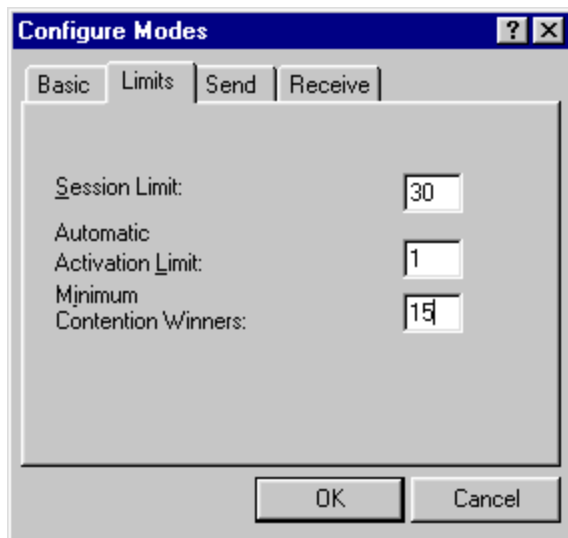
The MPTN Configuration dialog box is shown with the 'MPTN Advanced' tab selected. The 'Link Name' field contains 'CONN400'. The 'Link Type' section has three radio buttons: 'Mainframe', 'Peer', and 'AS/400', with 'AS/400' selected. The 'TCP/IP Domain Name' field is empty, and the 'Remote Host Name' field contains '012.345.678.901'. The 'Security' section has a checkbox labeled 'Use Host APPN Remote Location List' which is unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

5. Accept the defaults on the **MPTN Advanced Tab** (not shown), and then click the **OK** button to return to the RUMBA APPC Configuration window.
6. Click within the **Modes** pane, and then click the **Add** button.
7. The **Configure Modes** dialog box appears.



The Configure Modes dialog box is shown with the 'Basic' tab selected. The 'Mode Name' field contains 'IBMRDB'. 'OK' and 'Cancel' buttons are at the bottom right.

8. Type **IBMRDB** as the **Mode Name** in the text box. Select the **Limits** tab; enter the corresponding information. These values are compatible with the [IBMRDB](#) mode definition.

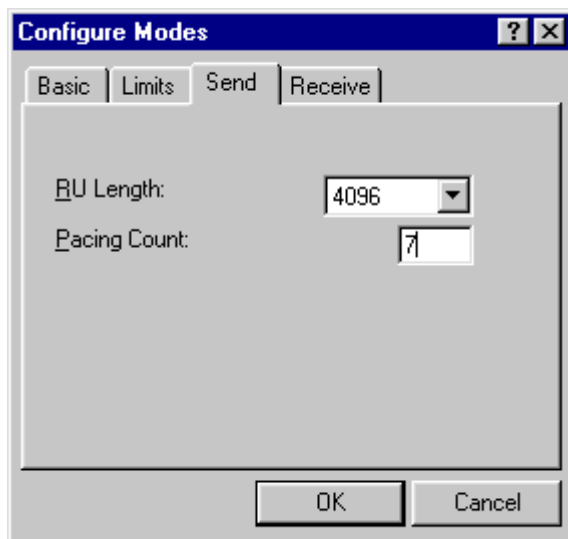


Session Limit = Maximum number of sessions

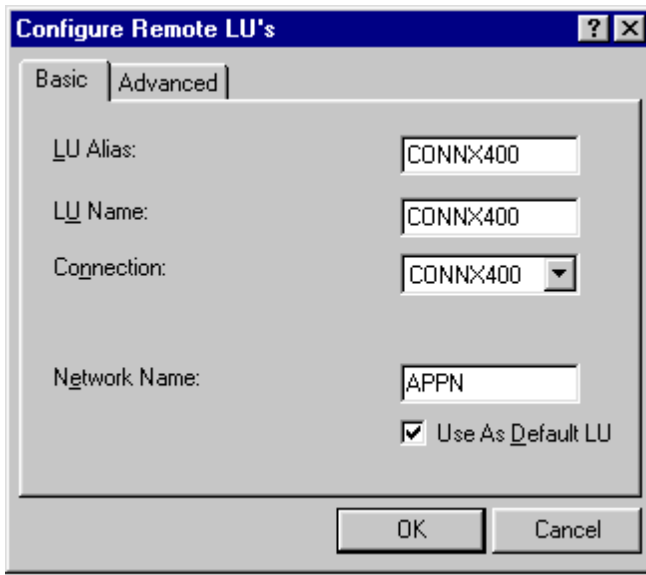
Automatic Activation Limit = Number of pre-established sessions

Minimum Contention Winners = Number of locally controlled sessions

9. Click the **Send** tab; and verify that the default RU length is set to **4096**, and that the Pacing Count is set to **7**.



10. Click the **Receive** tab; verify that the default RU length is set to **4096**, and that the Pacing Count is set to **7**. Click the **OK** button to return to the RUMBA APPC Configuration window.
11. Click within the **Remote LU's** pane, and then click the **Add** button. The **Configure Remote LU's** dialog box appears.



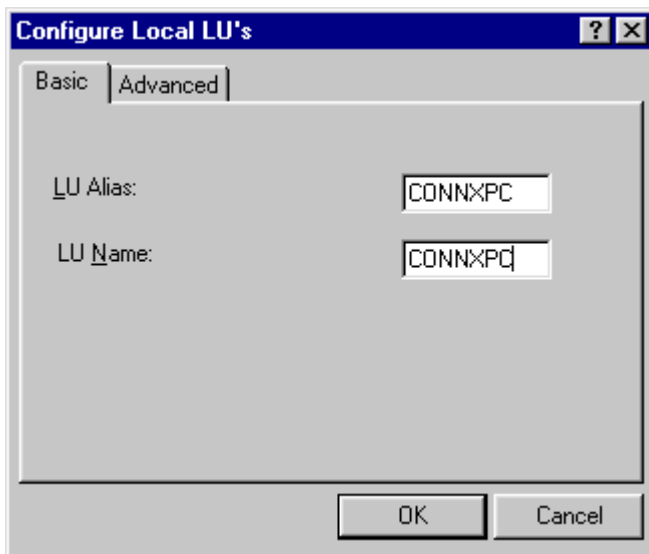
12. Select the **Basic** tab. In the **LU Alias** text box, type a symbolic 1- to 8-character LU alias. In the **LU Name** text box, type an LU name for the target AS/400 machine. Pick the previously defined CONNX400 connection from the list box.

In the **Network Name** text box, type the network name of the host, which can be displayed via DSPNETA command. The mapping from TCP/IP address to an APPC netid.luname is implemented in the `\\winnt\\system32\\drivers\\etc\\hosts` or `windows\\hosts` file, as follows:

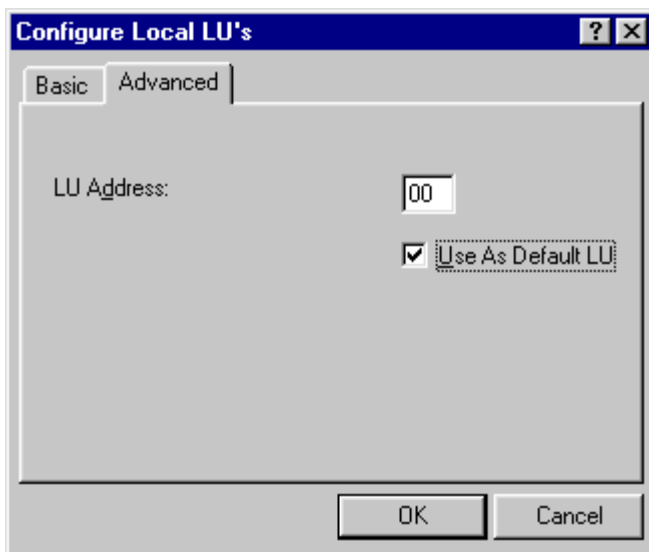
012.345.678.012 connx400.appn.sna.ibm.com #Anynet

Thus, the SNA netid.luname is **appn.connx400**, which should match the target system name as displayed in the OS/400 DSPNETA command output. The sna.ibm.com suffix is discarded by the RUMBA MPTN driver. If this entry is not present in your local client hosts file, it must exist in a TCP/IP domain name server on your network.

13. Click the **OK** button to return to the RUMBA APPC Configuration window.
14. Click within the **Local LU's** pane, and then click the **Add** button. The **Configure Local LU's** dialog box appears.
15. Select the **Basic** tab. For the purposes of this example, both the LU alias and the LU name are set to CONNXPC.



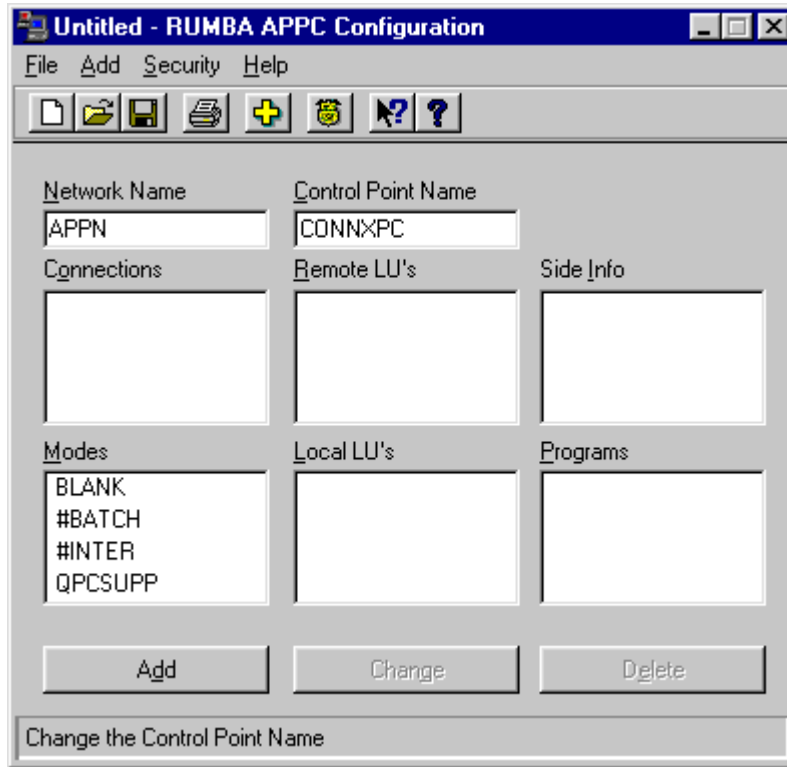
16. Select the **Advanced** tab. The LU address should be **00**. You can select the **Use As Default LU** option to define the CONNXP Local LU as the default for this configuration. Click the **OK** button to return to the RUMBA APPC Configuration window.



17. Select **Save** on the **File** menu in the RUMBA APPC Configuration window. You will need the information defined in this file when you connect via the CONNX CDD Import dialog box.

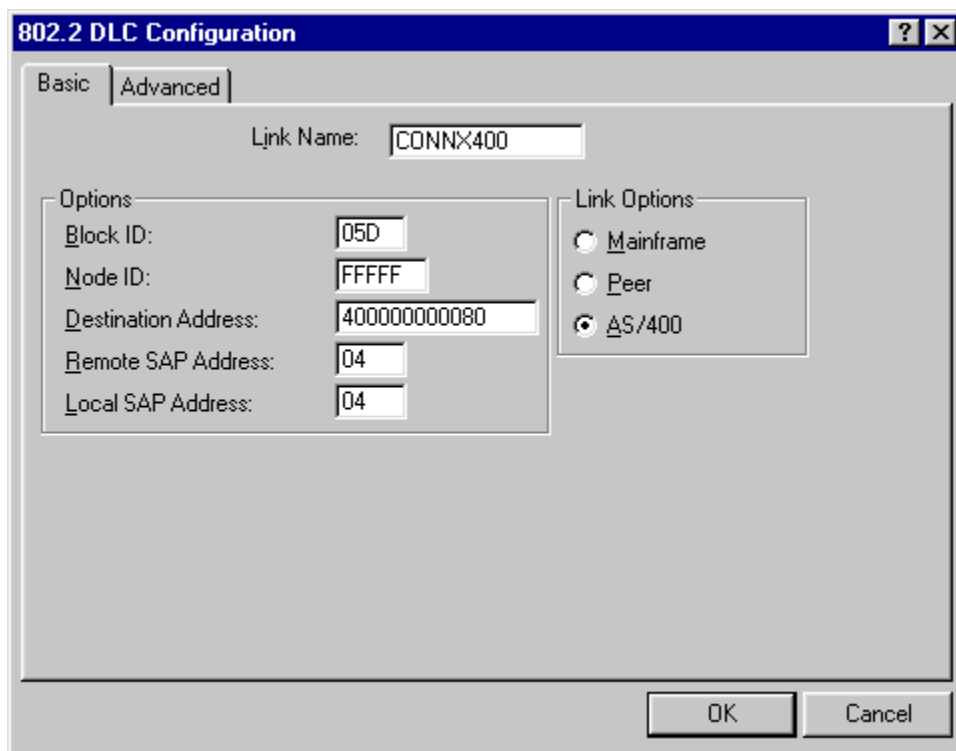
To set up 2- tier SNA client configuration for RUMBA IEEE 802.2 DLC

1. Select **Start**, point to **Programs**, and then select the **RUMBA APPC** configuration utility. The RUMBA APPC Configuration window appears. For this example, enter APPN as the network name.
2. Type the 1- to 8-character name of your client computer in the **Control Point Name** text box.



3. Click within the **Connections** pane, and then click the **Add** button. The **Select a Connection** dialog box appears. Select **802.2 DLC (Token Ring and Ethernet)**, and then click the **OK** button.

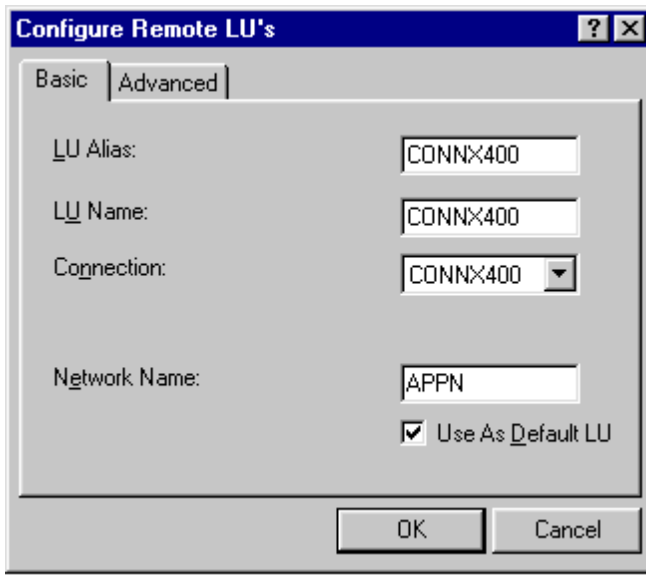
- The **802.2 DLC Configuration** dialog box appears.



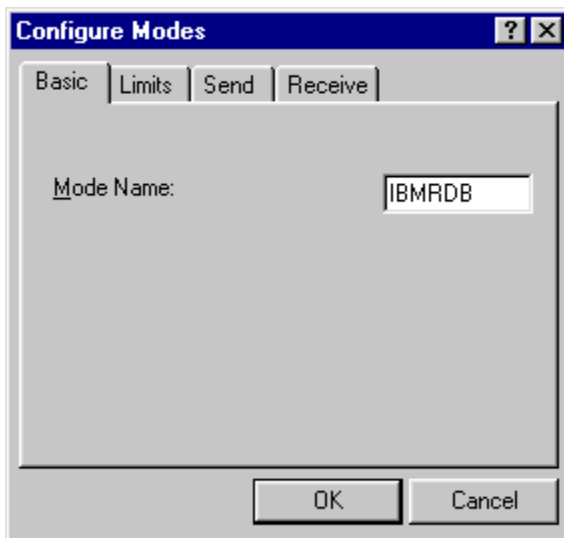
- Select the **Basic** tab. In the **Link Name** dialog box, type a user-defined 1- to 8-character link name. This example uses a link name of CONNX400. In the **Block ID** text box under **Options**, enter the block identification number, which should match the first 3 digits of the Exchange Identifier field, as displayed by entering the [WRKLIND](#) command.
- Enter the **Destination Address** of the host machine. The value of this field must match the LAN adapter address for the target system, which can be displayed via the OS/400 WRKLIND command. If you are not authorized to use this command, ask your OS/400 systems administrator for the target machine LAN adapter (MAC) address.

In the **Node ID** text box under **Options**, enter the node identification number, which should match the last 5 digits of the Exchange Identifier field, as displayed by entering the WRKLIND command.

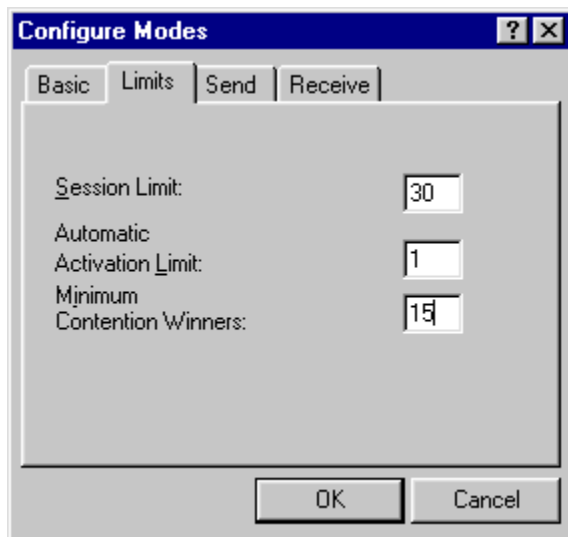
- Select the **Advanced** tab (not shown). Accept the default values, and then click the **OK** button to return to the RUMBA APPC Configuration window.
- Click within the **Remote LU's** pane, and then click the **Add** button. The **Configure Remote LU's** dialog box appears.



9. Select the **Basic** tab. In the **LU Alias** text box, type a symbolic 1- to 8-character LU alias. In the **LU Name** text box, type a 1- to 8-character LU name, which should match the local control point name field as displayed by the OS/400 DSPNETA command. Select the previously defined CONNX400 connection from the **Connection** list box.
10. In the **Network Name** text box, type a 1- to 8-character network name, which should match the Local Network ID field as displayed by the OS/400 [DSPNETA](#) command.
11. Thus, the SNA netid.luname is **appn.connx400**, which should match the target system name as displayed in the OS/400 DSPNETA command output.
12. Click the **OK** button to return to the RUMBA APPC Configuration window.
13. Click within the **Modes** pane, and then click the **Add** button.
14. The **Configure Modes** dialog box appears.



15. Enter [IBMRDB](#) in the **Mode Name** text box. Select the **Limits** tab; enter the corresponding information, which should match the information from the OS/400 DSPMODD command for mode IBMRDB.

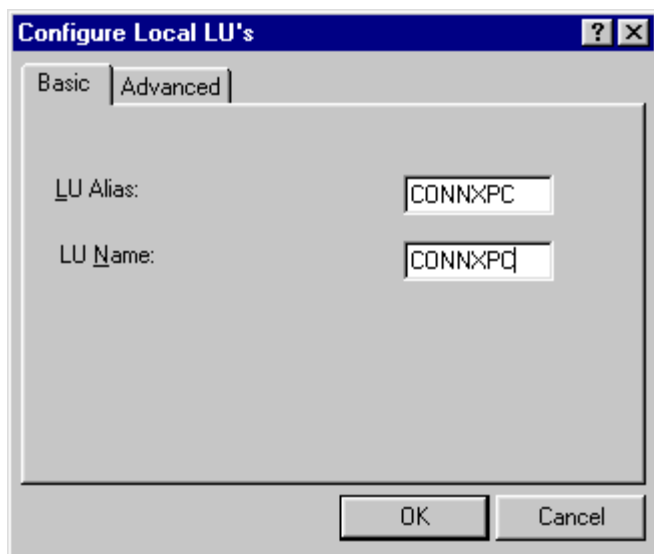


Session Limit = Maximum number of sessions

Automatic Activation Limit = Number of pre-established sessions

Minimum Contention Winners = Number of locally controlled sessions

16. Click the **Send** tab, and verify that the default RU length is set to **4096**, and that the Pacing Count is set to **7**.
17. Click the **Receive** tab, and verify that the default RU length is set to **4096**, and that the Pacing Count is set to **7**. Click the **OK** button to return to the RUMBA APPC Configuration window.
18. Click within the **Local LUs** pane, and then click the **Add** button. The **Configure Local LUs** dialog box appears.
19. Select the **Basic** tab. For purposes of this example, the client machine (Local LU name) is defined as CONNXP, and its LU alias is also CONNXP.



20. Select the **Advanced** tab. The default LU Address should be **00**. You can make this Local LU the default by selecting the **Use As Default LU** option. Click the **OK** button to return to the RUMBA APPC Configuration window.

21. Select **Save** on the **File** menu in the RUMBA APPC Configuration window. You will need the information defined in this file when you connect via the CONNX CDD Import dialog box.

CONNX and OS/390 and MVS/ESA Databases

Host Configuration

To define the RDBName to DB2 DDF

Prior to commencing host configuration for connection to OS/390 or MVS/ESA databases, you must first define the location name (RDBName) to the DB2 Distributed Data Facility (DDF).

DB2/mainframe: To define the RDBName to DB2 DDF

To support connections to mainframe DB2 products (DB2 for OS/390 or DB2 for MVS), the DB2 DDF host component must be configured and started. The DDF "listens for" or "catches" Distributed Relational Database Architecture (DRDA) requests originated by CONNX and other client software products which implement the DRDA Application Requester (AR) functionality. The first step is to define the RDBName or location to DDF. This can be done in at least two ways:

1. At install time, when the DB2 product is configured, the DSNTIPR installation panel (not shown) provides a prompt and an entry field for the location name, which is a 1- to 18-character alphanumeric string (16-character for DB2 for MVS/ESA).
2. After installation you can change or add the location name via an interactive TSO CLIST = DSNTINST.

Each of these methods updates the DDF record of the DB2 Bootstrap Dataset (BSDS), which provides installation parameters at DB2 startup time. The following screen shots depict the invocation of the DSNTINST CLIST and navigation to the SPF panel which displays the location (RDBName):

```

Session C - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help

INSTALL, UPDATE, AND MIGRATE DB2 - MAIN PANEL

====>
DSNT401I Warning - help is not installed
Check parameters and reenter to change:

1  INSTALL TYPE          ===> UPDATE      Install, Update, or Migrate
2  DATA SHARING         ===>             Yes, No, or blank for Update

Enter the following value for migration only:
3  DATA SET NAME(MEMBER) ===>

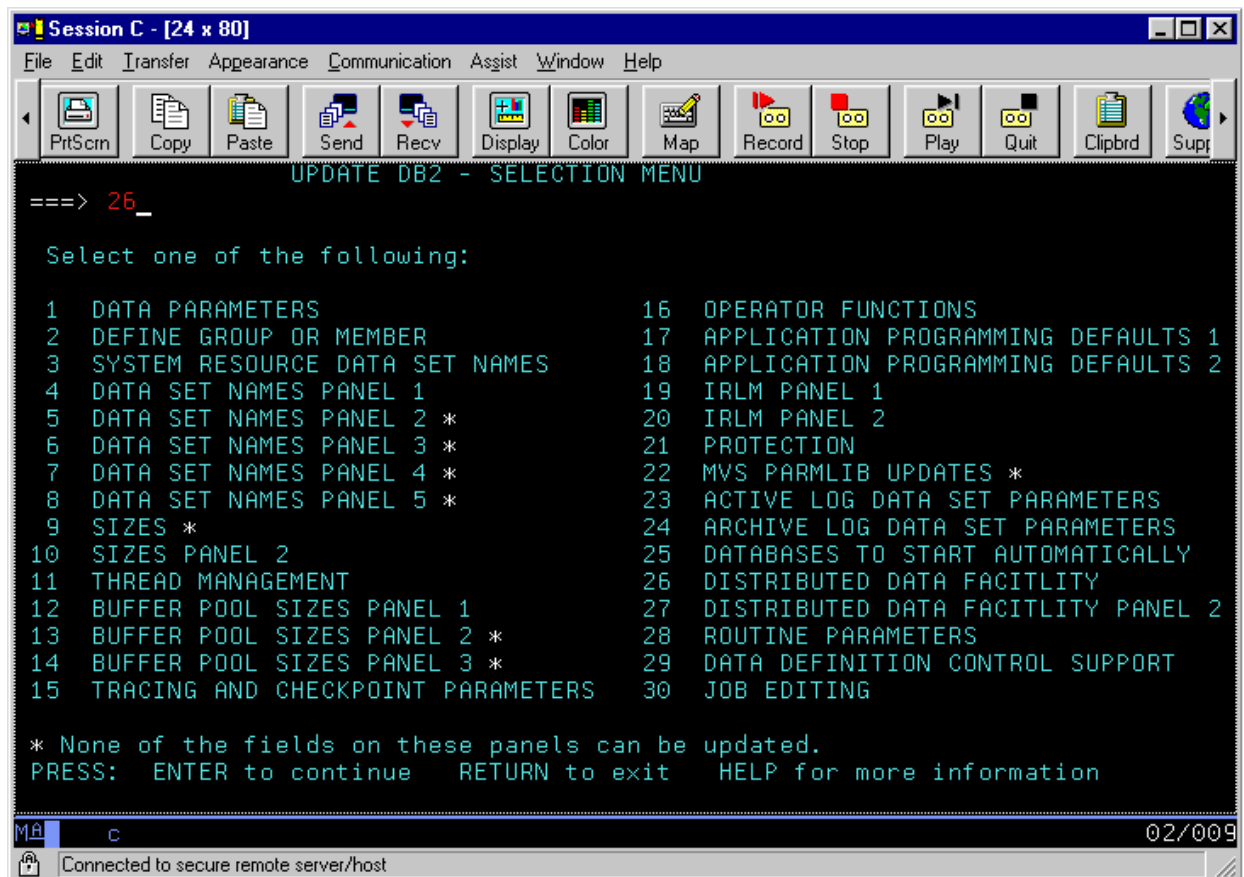
Enter name of your input data sets (SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNCLST):
4  PREFIX                ===> DSN610
5  SUFFIX                ===>

Enter to set or save panel values (by reading or writing the named members):
6  INPUT MEMBER NAME     ===> DSNTIDXA   Enter to read old panel values
7  OUTPUT MEMBER NAME    ===>           Enter to write new panel values

PRESS:  ENTER to continue  RETURN to exit  HELP for more information

MA c 02/007
Connected to secure remote server/host

```



To configure DDF for TCP/IP

Prior to commencing host configuration for connection to OS/390 or MVS/ESA databases, you must first define the location name (RDBName) to the DB2 Distributed Data Facility (DDF).

DB2/mainframe: To define the RDBName to DB2 DDF

To support connections to mainframe DB2 products (DB2 for OS/390 or DB2 for MVS), the DB2 DDF host component must be configured and started. The DDF "listens for" or "catches" Distributed Relational Database Architecture (DRDA) requests originated by CONNX and other client software products which implement the DRDA Application Requester (AR) functionality. The first step is to define the RDBName or location to DDF. This can be done in at least two ways:

1. At install time, when the DB2 product is configured, the DSNTIPR installation panel (not shown) provides a prompt and an entry field for the location name, which is a 1- to 18-character alphanumeric string (16-character for DB2 for MVS/ESA).
2. After installation you can change or add the location name via an interactive TSO CLIST = DSNTINST.

Each of these methods updates the DDF record of the DB2 Bootstrap Dataset (BSDS), which provides installation parameters at DB2 startup time. The following screen shots depict the invocation of the DSNTINST CLIST and navigation to the SPF panel which displays the location (RDBName):

```

Session C - [24 x 80]
File Edit Transfer Appearance Communication Assist Window Help
PrtScr Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipbrd Supr

INSTALL, UPDATE, AND MIGRATE DB2 - MAIN PANEL

===>
DSNT401I Warning - help is not installed
Check parameters and reenter to change:

1  INSTALL TYPE           ===> UPDATE      Install, Update, or Migrate
2  DATA SHARING          ===>             Yes, No, or blank for Update

Enter the following value for migration only:
3  DATA SET NAME(MEMBER) ===>

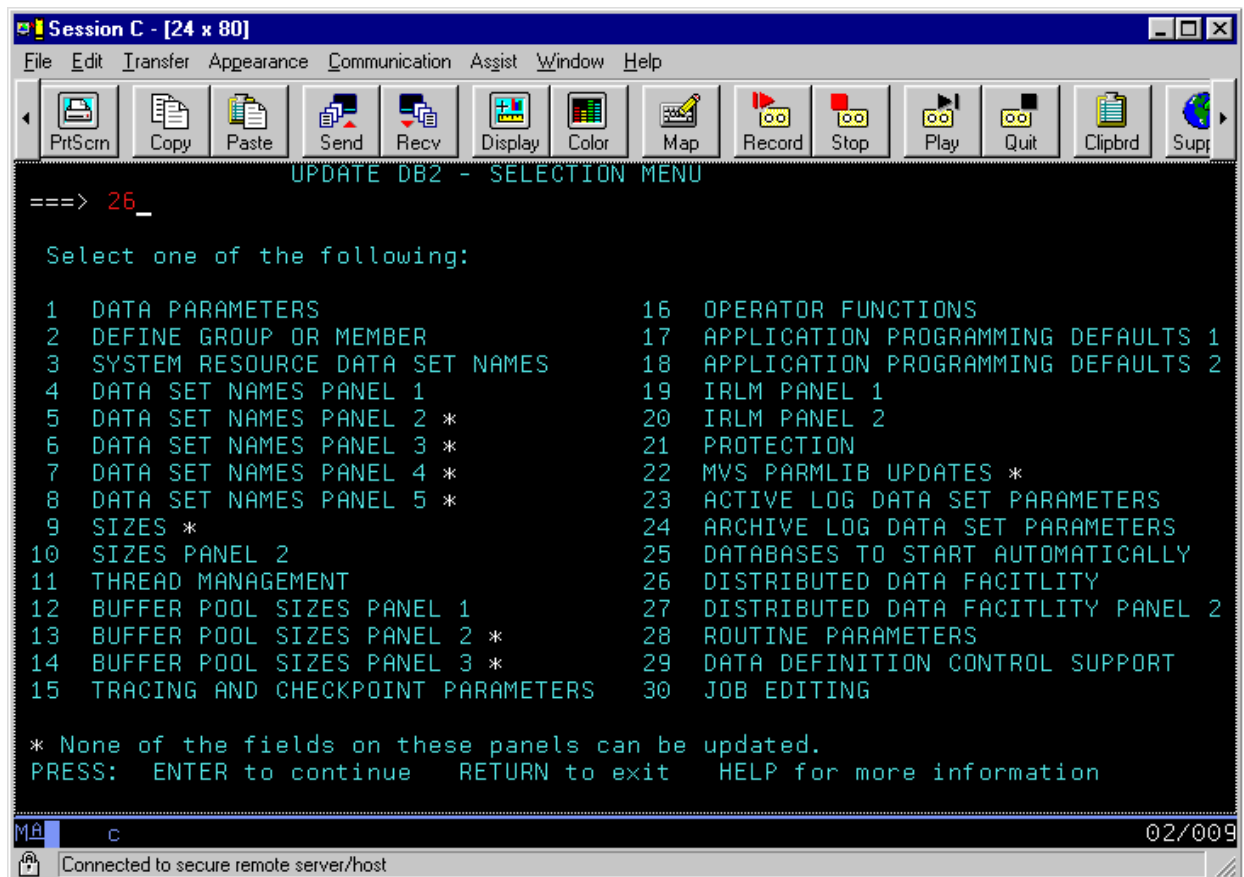
Enter name of your input data sets (SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNCLST):
4  PREFIX                 ===> DSN610
5  SUFFIX                 ===>

Enter to set or save panel values (by reading or writing the named members):
6  INPUT MEMBER NAME      ===> DSNTIDXA  Enter to read old panel values
7  OUTPUT MEMBER NAME     ===>           Enter to write new panel values

PRESS:  ENTER to continue  RETURN to exit  HELP for more information

MA  c 02/007
Connected to secure remote server/host

```



To configure VTAM for DRDA via SNA/LU 6.2

To enable DB2 to communicate using the DRDA datastream via APPC SNA/LU 6.2, several DDF VTAM parameters must be configured.

1. Define the DB2 system LU Name to DB2/390/MVS BSDS (bootstrap data set). The APPC logical unit which DDF uses to manage APPC SNA/LU 6.2 connections can be defined via the DSNTINST TSO CLIST; refer to the DB2 Network LUNAME prompt (option 37).
2. At startup, DDF reads the LU name defined in the DB2 BSDS and prepends the unique system VTAM net ID to form a fully qualified netid.luname, which is used to manage APPC LU 6.2 conversations (connections) between DRDA clients and the target database instance.
3. Define the VTAM APPLID. The LUNAME defined in the DB2 BSDS must correspond to a VTAM APPLID; a sample VTAM macro definition for the LU name LU1 appears below:

```

DB2 APPL      VBUILD TYPE  =  APPL
LU1           APPL APPC    =  YES
              AUTH         =  (ACQ)
              AUTOSSES     =  1,
              DSESLIM      =  20,
              DMINWNL      =  10,
              DMINWNR      =  10,
              SECACPT       =  ALREADYV,
              EAS          =  509,
              MODETAB      =  DB2MODES.
              PARSESS      =  YES,
              SRBEXIT       =  YES,
              VPACING       =  2
  
```

The important parameters above are the APPL ID = LU1, which must match the value of the LU name entered in the DB2 BSDS. Additionally, setting SECACPT = ALREADYV provides for maximum flexibility when validating DB2 userid/password security at connect time. When SECAPT=ALREADYV is configured, the following connection options are available.

Connection Options

APPC Security	CONNX Option	Values sent from client to host
Same	UserID	UserID only
Program	UserID & Password	User ID and password
None	None	None, but DB2 for OS/390 and MVS reject this option.

4. Define the VTAM Mode Name (IBMRDB). The VTAM APPLID definition above references a VTAM mode table = DB2Modes; within this mode table, mode name IBMRDB or a custom mode name must be defined. The IBMRDB mode name is the IBM-recommended default for managing DRDA datastream traffic via APPC LU 6.2.

IBMRDB

MODEENT	LOGMODE=IBMRDB,	DRDA DEFAULT MODE
	TYPE = 1,	NEGOTIABLE BIND
	PSNDPAC=X'00',	PRIMARY SEND PACING COUNT
	SSNDPAC=X'02',	SECONDARY SEND PACING COUNT
	SRCVPAC=X'00',	SECONDARY RECEIVE PACING COUNT
	RUSIZES=X'8989',	RUSIZES IN -4K OUT-4K
	FMPROF=X'13',	LU6.2 FM PROFILE
	TSPROF=X'07',	LU6.2 TS PROFILE
	PRIPROT=X'BO',	LU6.2 PRIMARY PROTOCOLS
	SECPROT=X'BO',	LU6.2 SECONDARY PROTOCOLS
	COMPROT=X'50A5',	LU6.2 COMMON PROTOCOLS
	PSERVIC=X'06020000000000000000102F00',	LU6.2 LU TYPE
APPNCOX=#CONNECT		

Configuring MPTN-specific VTAM Information on the Host

To create an SNATCP macro

The following includes the definition for using SNA over TCP/IP:

STCPIP	VBUILD TYPE	=	TCP
	DNSUFFIX	=	CONNX.DB2390.com,
	TCP	=	50,
	PORT	=	397,
	STCPGI GROUP I STATUS	=	ACTIVE,
	STCPL1 LINE STATUS	=	ACTIVE,
	STCPP1 PU STATUS	=	ACTIVE,
	NETID	=	CONNXDB2

DNSUFFIX is the TCP/IP domain name suffix for symbolic TCP/IP addresses. Full MPTN/TCP/IP addresses are of the form Luname.netid.suffix, where suffix can be a three-part name as defined above.

Note that port 397 is reserved for MPTN traffic and that NetID is the VTAM Net ID to which the target host belongs. CONNXDB2 is a sample Net ID, and CONNX.DB2390.COM is a sample DNS suffix.

To verify correct IP address and host name

- Confirm that the correct IP address and host name are defined in the TCPIP.HOSTS.LOCAL sequential dataset. The following is a sample hosts file:

HOSTS.LOCAL

COPYRIGHT = NONE

The format of this file is documented in RFC 952, "DoD Internet Host Table Specification."

The format for entries is as follows:

NET:ADDR:NETNAME:

GATEres:ADDR,ALT-ADDR:HOSTNAME:CPUTYPE;;OPSYS PROTOCOLS:

HOST:ADDR,ALT-ADDR:HOSTNAME,NICKNAME:CPUTYPE OPSYS:PROTOCO

where

ADDR,ALT-ADDR=IP address in decimal, e.g., 012.345.678.901

HOSTNAME,NICKNAME=the fully qualified host name and any nicknames

CPUTYPE=machine type (PDP-11/70, VAX-11/780, IBM-3090, C/30, etc.)

OPSYS=operating system (UNIX, TOPS20, TENES, VM/SP, etc.)

PROTOCOLS=transport/service (TCP/TELNET, TCP/FTP, etc.)

:(colon)=field delimiter

::(2 colons)=null field

Notes:

CPUTYPE, OPSYS, and PROTOCOLS are optional fields.

HOST:012.345.678.901:CONNxDB2.CONNX.DB2390.COM, CONNXDB2

DB2 Communications Database Setup

Setup

DB2 for OS/390 and DB2 for MVS manage inbound and outbound security via a set of tables called the communications database (CDB). Inbound security validation for APPC/LU 6.2 DRDA clients is managed by the SYSIBM.SYSLUNAMES (DB2/MVS) or SYSIBM.LUNAMES (DB2/OS390) table. To implement the simplest security model, enter a blank row in this table:

```
INSERT into SYSIBM.LUNAMES (LUNAME) VALUES ( ' ' );
```

When table SYSIBM.(SYS)LUNAMES contains a blank row, DB2 forwards the userid/password combination sent by the DRDA client (CONNX) to the system-installed security component, such as RACF, ACF/VTAM, or TopSecret.

For TCP/IP-based connections, the SYSIBM.(SYS)LUNAMES tables is bypassed, so it need not be populated.

This table represents sample output from SELECT * from SYSIBM.LUNAMES:

SELECT sample output

APPC Security	CONNX Option	Values sent from client to host
Same	UserID	UserID only.
Program	UserID & Password	User ID and password.
None	None	None, but DB2 for OS/390 and MVS reject this option.

APPC Connection Scenarios - 3-Tier Communications Server Configuration

To connect to a DB2/MVS or DB2 for OS/390

If you plan to connect to a DB2/MVS or DB2 for OS/390 region via APPC/LU 6.2 over AnyNet/MPTN, the following is a sample configuration scenario using the IBM Communications Server 6.0.2 for Windows.

1. Define the node. Start the IBM Communications Server SNA Node Configuration program. Select **Scenarios+AnyNet SNA over TCP/IP Gateway** from the main menu. Click the **Node** element in the **Definition Hierarchy by Function** tree view, and then click the **Create** button. Define an example network node=APPN.CONNXDB2.

Define the Node

Basic | Advanced | DLU Requester

Control Point (CP)

Fully qualified CP name:

APPN . CONNXDB2

CP alias:

CONNXGW

Local Node ID

Block ID: 05D Physical Unit ID: 00000

Node Type

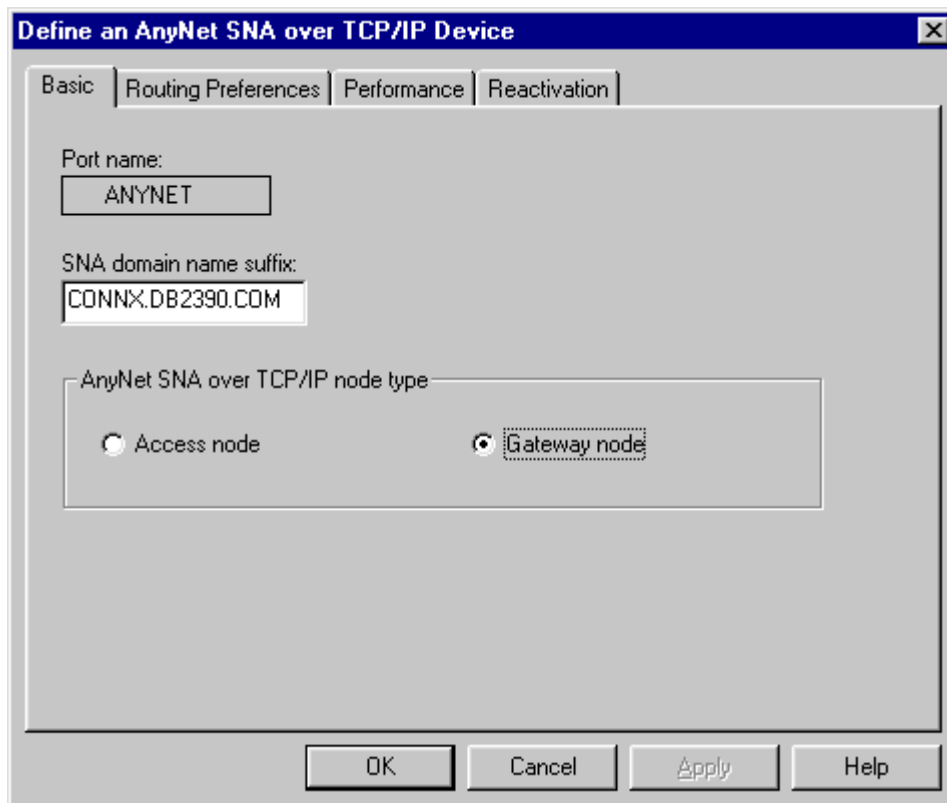
☐ End Node

☒ Network Node

☐ Branch Extender Node

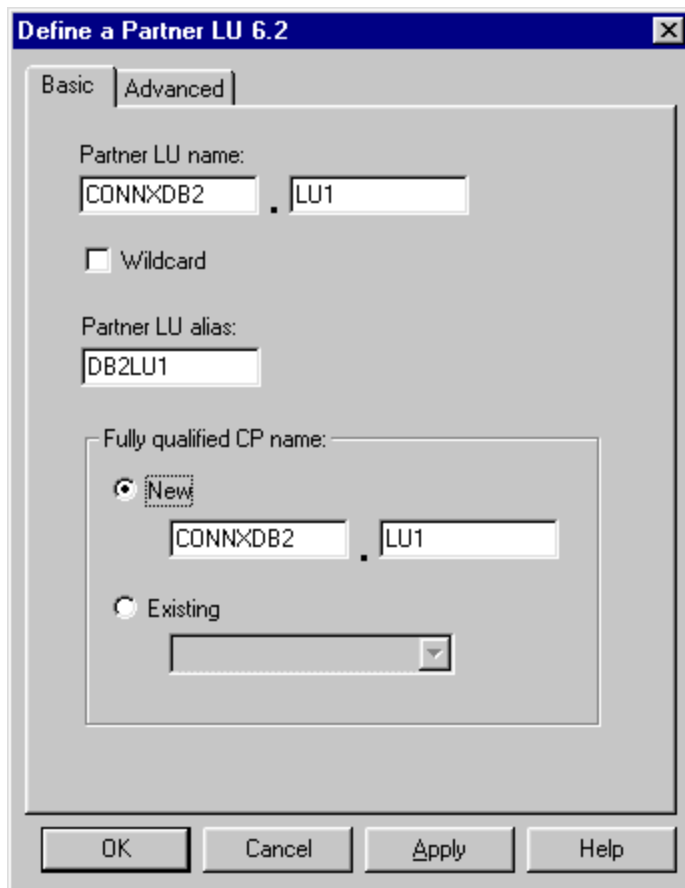
OK Cancel Apply Help

2. Define an **AnyNet SNA/IP** gateway device. Click the **Devices** element in the **Definition Hierarchy by Function** directory, and then click the **Create** button. Select **AnyNet SNA/IP** from the **Device Type** dialog box (not shown).
3. The **Define an AnyNet SNA over TCP/IP Device** dialog box appears. For this example, the SNA domain name suffix is set to CONNX.DB2390.COM. Accept the defaults for the **Routing Preferences**, **Performance**, and **Reactivation** dialog box tabs.

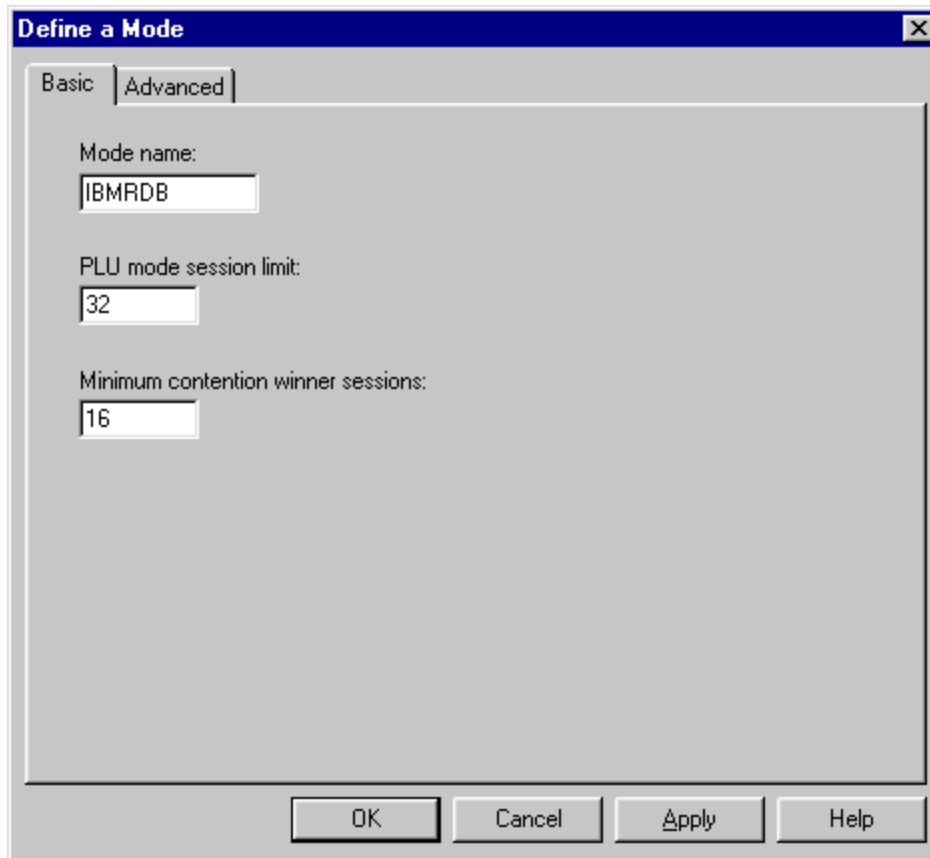


The dialog box is titled "Define an AnyNet SNA over TCP/IP Device". It has four tabs: "Basic", "Routing Preferences", "Performance", and "Reactivation". The "Basic" tab is selected. Inside the dialog, there are two text input fields: "Port name:" with the value "ANYNET" and "SNA domain name suffix:" with the value "CONNX.DB2390.COM". Below these fields is a section titled "AnyNet SNA over TCP/IP node type" containing two radio buttons: "Access node" (unselected) and "Gateway node" (selected). At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

4. Define a **Partner LU**. Click the **CPI-C** and **APPC** element in the **Definition Hierarchy by Function** directory, click on the **Partner LU 6.2 LUs** element, and then click the **Create** button.
5. The **Define a Partner LU 6.2** dialog box appears. For this example, define a Partner LU = CONNXDB2.LU1 with a partner LU alias = DB2LU1. The NetID.LU name must match the target host system VTAM Net ID and the VTAM APPLID defined to the target DB2 DDF region, as documented in [To configure VTAM for DRDA via SNA/LU 6.2](#).



6. Define the **IBMRDB** mode name. Click the **CPI-C** and **APPC** element in the **Definition Hierarchy by Function** directory, click the **Modes** element, and then click the **Create** button.
7. The **Define a Mode** dialog box appears. For this example, the mode name is **IBMRDB**, which is preconfigured at the host and optimized for DRDA traffic.



The image shows a Windows-style dialog box titled "Define a Mode". It has a blue title bar with a close button (X) in the top right corner. Below the title bar, there are two tabs: "Basic" and "Advanced". The "Basic" tab is currently selected. Inside the dialog, there are three text input fields. The first is labeled "Mode name:" and contains the text "IBMRDB". The second is labeled "PLU mode session limit:" and contains the number "32". The third is labeled "Minimum contention winner sessions:" and contains the number "16". At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Define a Mode

Basic | Advanced

Mode name:
IBMRDB

PLU mode session limit:
32

Minimum contention winner sessions:
16

OK Cancel Apply Help

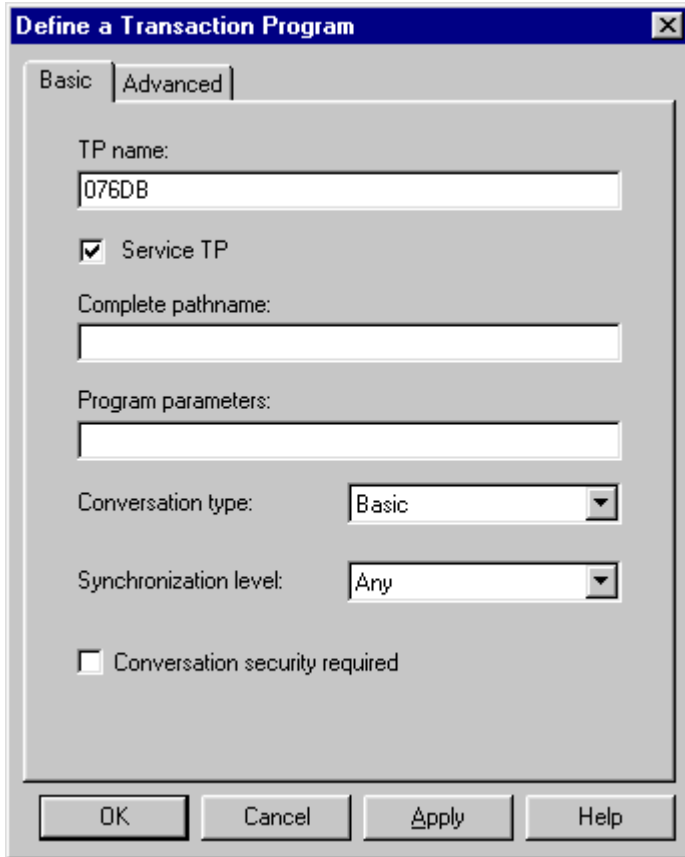
8. Select the **Advanced** tab.

The screenshot shows the 'Define a Mode' dialog box with the 'Advanced' tab selected. The dialog has a title bar with a close button. Inside, there are two tabs: 'Basic' and 'Advanced'. The 'Advanced' tab contains the following settings:

- Maximum negotiable session limit: 128
- Receive pacing window size: 7
- Auto activate sessions: 0
- Class of Service name: #CONNECT (dropdown menu)
- ☒ Use default RU size
 - Maximum RU size: 4096
- ☐ Encryption support
- ☐ Compression support
 - Maximum compression level for outbound data: <NONE> (dropdown menu)
 - Maximum compression level for inbound data: <NONE> (dropdown menu)

At the bottom of the dialog are four buttons: OK, Cancel, Apply, and Help.

9. Define the **Transaction Program** name. Click the **CPI-C** and **APPC** element in the **Definition Hierarchy by Function** directory, select **Transaction Programs**, and then click the **Create** button.
10. The **Define a Transaction Program** dialog box appears. The default DRDA Application Server APPC transaction program is = 0x07F6C4C2, which is entered as 076DB.



The image shows a Windows-style dialog box titled "Define a Transaction Program". It has two tabs: "Basic" (selected) and "Advanced". The "Basic" tab contains the following fields and controls:

- TP name:** A text box containing "076DB".
- Service TP:** A checked checkbox.
- Complete pathname:** An empty text box.
- Program parameters:** An empty text box.
- Conversation type:** A dropdown menu with "Basic" selected.
- Synchronization level:** A dropdown menu with "Any" selected.
- Conversation security required:** An unchecked checkbox.

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

11. Save the configuration file and set it as the default. For this example, the following entries must be added to the **winnt\system32\drivers\etc\hosts** file on the same server:

```
123.456.789.12 LU1.CONNXDB2.CONNX.DB2390.COM
```

```
123.456.789.12 LU1
```

where

123.456.789.120 is a sample dotted TCP/IP address, which must match the TCP/IP address defined to the target DB2 DDF region.

LU1.CONNXDB2.CONNX.DB2390.COM is the full symbolic TCP/IP address of the target host, in luname.netid.DNS-suffix order. The default AnyNet/MPTN DNS suffix is sna.ibm.com; for the purposes of this example, the DNS suffix was changed to CONNX.DB2390.COM.

The IBM Communications Server MPTN driver forms a symbolic name in luname.netid.DNS-suffix order from its configuration, and looks up the dotted TCP/IP address in the hosts file. If the entry is not in the local hosts file, it is resolved at the TCP/IP domain name server(s).

To connect to the target DB2 region via CONNX

1. Start the **CONNX CDD Import** application, enter the target **RDBName (DDF location)**, and then enter a user name and password.
2. Choose the **SNA** option, and then click the **SNA Settings** tab.
3. Select **IBM** as the vendor stack, and then enter the mode name = **IBMRDB** and either the full netid.luname (Remote Net ID and Remote LU) = **CONNXDB2.LU1** or the LU alias = **DB2LU1**.
4. CONNX loads the IBM Communications Server client, which connects to the IBM Communications Server, and then to the target host.

DB2 Bibliography and Reference Sources

References and Sources

To implement more restrictive security models via the DB2 OS/390 or MVS CDB, you can consult the following IBM publications, which can be ordered online at <http://www.redbooks.ibm.com/>:

Janacek, Calene, lead author. *The Universal Connectivity Guide to DB2*, Form # SG24-4894, IBM Redbook, International Business Machines, Inc., 1998.

Distributed Relational Database Architecture Connectivity Guide, Form # SC26-4783-03, International Business Machines, Inc.

The book has also been published by Prentice Hall, Inc., ISBN# 0-13-398306-4.

For further reference, consult the following publication:

Microsoft ODBC 3.0 Software Development Kit and Programmer's Reference. Edited by Casey D. Doyle, Maureen Williams Zimmerman, and Zephan Schroeder. Redmond, Washington: Microsoft Press, 1997.

Step 9 - Post-Installation Checklist

General Information Regarding the Post-Installation Checklist

You should complete a Post-Installation Checklist for each database you install.

If you are upgrading CONNX from an earlier version, complete the following checklist:

[Post-Installation Checklist - CONNX Upgrades.](#)

As you complete each Post-Installation Checklist, check off the corresponding item in the table below.

Post-Installation Checklist

Check here when complete	Item
	1. Completed Post-Installation Checklist - DataFlex
	2. Completed Post-Installation Checklist - POWERflex
	3. Completed Post-Installation Checklist - DB2
	4. Completed Post-Installation Checklist - Oracle
	5. Completed Post-Installation Checklist - OLE DB- and ODBC-compliant data sources
	6. Completed Post-Installation Checklist - Sybase data sources.
	7. Completed Post-Installation Checklist - Informix data sources
	8. Completed Post-Installation Checklist - C-ISAM, DISAM, or Micro Focus data sources.
	9. Completed Post-Installation Checklist - OpenVMS Server
	10. Completed Post-Installation Checklist - IBM Mainframe-compatible server.
	11. Completed Post-Installation Checklist - CONNX Upgrades
	12. Completed Post-Installation Checklist - CONNX JDBC Server Post-Installation Checklist - CONNX JDBC Server

Post-Installation Checklist - C-ISAM, DISAM, or Micro Focus

The following table provides a checklist of the post-installation operations you must perform before C-ISAM installation is complete. This checklist is not required for new installations. After each operation is finished, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-Installation Checklist - C-ISAM Server

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must be restored. To restore the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client procedure.
	2. Establish a connection with the C-ISAM test data by going to the Start menu, selecting CONNX8, selecting Build Sample Databases, and then selecting Build CISAM Sample Data. For a procedure on how to test the connection, see Testing a Database Connection .

Post-Installation Checklist - CONNX JDBC Server

The following table provides a checklist of the post-installation operations you must conduct before the OpenVMS installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure.

Post-Installation Checklist – JDBC Server

Check here when complete	Item
	1. Review CONNX JDBC information in Step 4, Install CONNX JDBC Server and Client.
	2. Verify that a CONNX JDBC Server is running. Refer to CONNX JDBC Driver in the CONNX User Reference Guide for installation instructions.
	3. Verify that a router is installed and running, if required. Refer to CONNX JDBC Driver in the CONNX User Reference Guide for installation instructions.
	4. Using the test data and the CONNX JDBC Sample Application supplied with the CONNX JDBC installation, establish a connection with the JDBC server for each installed database, for example, DB2, DataFlex, or Oracle Rdb.

Related Topics

» JDBC Resources

» [Activating the CONNX JDBC Router](#)

» [Connecting to CONNX JDBC Server](#)

Post-Installation Checklist - CONNX Upgrades

The following table provides a checklist of the post-installation operations you must perform before the CONNX upgrade installation is complete. This checklist is not required for new installations. After each operation is finished, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Hints for Installing CONNX Upgrades

To Version	From Version:	Hints:
Latest CONNX version	CONNX 6.0 or higher	<ol style="list-style-type: none"> 1. Convert your CONNX Data Dictionaries (CDDs) into the latest revision format. To do this, complete the Converting Data Dictionaries procedure. 2. Activate your existing license files. See the procedures described in License Activation .
Latest CONNX version	CONNX 5.0 to CONNX 5.9	<ol style="list-style-type: none"> 1. Convert your CONNX Data Dictionaries (CDDs) into the latest revision format. To do this, complete the Converting Data Dictionaries procedure. 2. Install all-new license files you received with the upgrade. 3. Activate your existing license files. See the procedures described in License Activation .

Post-Installation Checklist - DataFlex

The table below provides a checklist of the Post-Installation operations you must conduct before the DataFlex installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-installation Checklist – DataFlex

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the DataFlex test data. For a procedure on how to test the connection, see Testing a Database Connection .
	3. To initialize CONNX to handle DataFlex dates for 2-digit year or 4-digit years, modify the C:\Windows\CONNX.INI file. For a procedure on how to modify the CONNX.INI file, see Modifying the CONNX Registry File Settings for DataFlex Year Digits .
	To customize how CONNX is initialized, see CONNX Registry File Settings and CONNX CDD Registry File Settings in the CONNX User Reference online help files.

Related Topic

 [License Activation](#)

Modifying the CONNX Registry File Settings for DataFlex Year Digits

From Item 3 of [Post-Installation Checklist - DataFlex](#), complete the following procedure to modify the CONNX file settings to incorporate DataFlex 2-digit or 4-digit years.

To modify the CONNX.INI file for DataFlex Year Digits

1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Configuration Manager**. The CONNX Configuration Manager window appears.
2. On the **CONNX Settings** tab, locate Software\CONNX Solutions\CONNX\DATAFLEX on the **Current Key** drop-down list.
3. In the **Selected Value Name** text box, type **2DigitYear**
4. In the **Selected Value Data** text box, type either **1** or **0**, depending on whether you want your DataFlex query results to include 2Digit Years, as described in the following:

[HKEY_LOCAL_MACHINE\SOFTWARE\CONNX SOLUTIONS\CONNX\DATAFLEX]

2DigitYears=1 <==True: 2-digit years selected

—or—

[HKEY_LOCAL_MACHINE\SOFTWARE\CONNX SOLUTIONS\CONNX\DATAFLEX]

2DigitYears=0 <==False: 4-digit years selected.

5. Click the **Save** button to save the file settings, and then click the **Close** button.

Related Topic

 [Post-Installation Checklist - DataFlex](#)

 [License Activation](#)

Post-Installation Checklist - DB2

The following table provides a checklist of the post-installation operations you must conduct before the DB2 installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-Installation Checklist – DB2

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 – Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the DB2 test data. For a procedure on how to test the connection, see Testing a Database Connection .
	To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.

Related Topic

 [License Activation](#)

Post-Installation Checklist - IBM Mainframe-compatible Server

The table below provides a checklist of the post-installation operations you must conduct before the installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-Installation Checklist – IBM Mainframe-compatible Server

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 – Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the VSAM test data. For a procedure on how to test the connection, see Testing a Database Connection .
	To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.

Post-Installation Checklist - Adabas SQL Gateway (CONNX for Adabas)

The table below provides a checklist of the post-installation operations you must conduct before the installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-Installation Checklist - Adabas SQL Gateway (CONNX for Adabas)

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the VSAM test data. For a procedure on how to test the connection, see Testing a Database Connection .
	To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.

Post-Installation Checklist - Informix data sources

The following table provides a checklist of the post-installation operations you must conduct before the Informix data source installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure.

Post-Installation Checklist - Informix Data Sources

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the Informix test data. For a procedure on how to test the connection, see Testing a Database Connection .
	3. To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.
	4. To properly configure the environment, define the following environment variables:: CLIENT_LOCALE=EN_US.CP1252 DB_LOCALE=EN_US.8859-1 INFORMIXDIR=/usr/local/informix Please consult your Informix documentation for information regarding specific values for your operating system
	5. For Informix on HPUX , run-time libraries iclit09b.sl and igo4a304.sl must be accessed from the \$CNXDIR /lib32 directory where CNXDIR is the location of the CONNX Client install directory. You can either copy the libraries into that location or create a symbolic link from their installed location(s) to CNXDIR.

Post-Installation Checklist - Oracle

The following table provides a checklist of the post-installation operations you must conduct before the Oracle installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Note: For Unix and Linux, the Oracle Client is installed as part of the CONNX installation. When installing CONNX on a Windows system, the Oracle Client must be downloaded from Oracle and installed separately.

Post-Installation Checklist – Oracle

Check here when complete	Item
	<p>1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers.</p> <p>Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased.</p> <p>This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 – Install CONNX Administrator and CONNX Client.</p>
	<p>2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the Oracle test data. For a procedure on how to test the connection, see Testing a Database Connection.</p>
	<p>3. To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.</p>
	<p>4. To properly configure the environment for Unix and Linux, define the following environment variables::</p> <p>ORACLE_HOME=\$CNXDIR/lib32 (Where CNXDIR is the CONNX Client install path)</p> <p>NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1</p> <p>For Windows, please download the Oracle Client from Oracle and install it according to the Oracle documentation.</p> <p>Please consult your Oracle documentation for information regarding specific values for your operating system</p>
	<p>5. For Unix and Linux, place a valid copy of tnsnames.ora into the directory defined by \$ORACLE_HOME. Please consult your Oracle documentation for information regarding the format of the tnsnames.ora configuration file.</p> <p>For Windows, please consult your Oracle documentation for</p>

	information regarding the location and format of the tnsnames.ora configuration file.
	<p>6. (Unix and Linux only) \$CNXDIR/lib32 must be the first entry on the dynamic library search path. If necessary, add or move the \$CNXDIR/lib32 path to the beginning of the LD_LIBRARY_PATH or equivalent environment variable search path.</p> <p>Your operating system guide will tell you how to set the dynamic library search path.</p>

Related Topic

 [License Activation](#)

Post-Installation Checklist - POWERflex

The following table provides a checklist of the Post-Installation operations you must conduct before the POWERflex installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-installation Checklist – POWERflex

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the POWERflex test data. For a procedure on how to test the connection, see Testing a Database Connection .
	3. To initialize CONNX to handle POWERflex dates for 2-digit year or 4-digit years, modify the registry file settings. For a procedure on how to modify the CONNX.INI file, see Modifying the CONNX Registry File Settings for DataFlex Year Digits .
	To customize how CONNX is initialized, see CONNX Registry File Settings and CONNXCDD Registry File Settings File Settings in the CONNX User Reference online help files.

Post-Installation Checklist – OLE DB- and ODBC-Compliant Data Sources

The following table provides a checklist of the post-installation operations you must conduct before the OLE-DB or ODBC-compliant data source installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure.

Post-Installation Checklist - OLE DB- and ODBC-compliant data sources

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must be restored. To restore the erased CONNX32 DSN, repeat " Step 2: Install CONNX Administrator and CONNX Client ".
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the OLE DB or ODBC test data. For a procedure on how to test the connection, see Testing a Database Connection .
	To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help file.

Post-Installation Checklist - OpenVMS Server

The following table provides a checklist of the post-installation operations you must conduct before the OpenVMS installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure, which you can reference as needed.

Post-Installation Checklist – OpenVMS Server

Check here when complete	Item
	<p>1. Review CONNX OpenVMS Installation Procedure. If you selected Shared Images and the OpenVMS server location, then you must edit the file for your site-specific startup command procedure.</p> <p>The file name depends on the version of OpenVMS or the VSM you are using: File for OpenVMS (any version).</p> <p>SYS\$MANAGER:SYSTARTUP_VMS.COM</p> <p>File for VMS versions 5.3 to 5.5: SYS\$MANAGER.SYSTARTUP_V5.COM</p> <p>The following lines are added to start CONNX when the OpenVMS server starts or restarts:</p> <pre> DEFINE /SYSTEM CNXRMSE0 SYS\$COMMON:[CONNX]CNXRMSE0.COM DEFINE /SYSTEM CNXRDBE0 SYS\$COMMON:[CONNX]CNXRDBE0.COM DEFINE /SYSTEM CNXDBME0 SYS\$COMMON:[CONNX]CNXDBME0.COM DEFINE /SYSTEM CNXSVR SYS\$COMMON:[CONNX]CNXSVR.COM DEFINE /SYSTEM CNXDIR SYS\$COMMON:[CONNX] @CNXSVR START </pre> <p>For a procedure on how to incorporate these lines, see Editing the OpenVMS Site-Specific Startup Command.</p>
	<p>2. Review the minimum quotas for the OpenVMS accounts that will use CONNX. The recommended minimum quotas are as follows:</p> <pre> Network Access: No Access Restrictions Max jobs: 0 Prclm: 8 BIOlm: 512 DIOlm: 512 Enqlm: 2048 Bytlm: 82000 JTQuota: 4096 Wsdef: 1024 Wsquo: 4096 Wsextant: 16348 Pgflquo: 82000 </pre>

	For a procedure on how to review these quotas, see Reviewing the Minimum OpenVMS Quotas Procedure .
	3. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 – Install CONNX Administrator and CONNX Client .
	4. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the OpenVMS server for each installed database, for example, RMS, RDB, and DBMS. For a procedure on how to test the connection, see Testing a Database Connection .
	5. To customize how CONNX is installed, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help file.

Related Topic

 License Upgrade Registration - OpenVMS

Editing the OpenVMS Site-Specific Startup Command

From Item 1 in the [OpenVMS Server Post-Installation checklist](#) complete the following procedure to edit the OpenVMS Site-Specific Startup Command Procedure.

To edit the OpenVMS Site-Specific Startup Command Procedure

1. On your CONNX administrator computer, start your terminal emulator, select the desired OpenVMS server, and type the appropriate user name and password. Proceed to a DCL prompt.
2. To edit the Site-Specific Startup Command Procedure, type the following:
`edit/edit SYS$MANAGER:SYSTARTUP_VMS.COM`
 and then press **<Enter>**. (VMS Version 5.3 to 5.5 users, type `SYSTARTUP_V5.COM` and then press **<Enter>**.) The first line of SYSTARTUP and the * prompt are displayed.
3. After the * prompt, type `1:END` which takes you to the end of the file.
4. Observe the number of the line with the EXIT command, which is at the end or near the end of the file. For example:

```
273 $ EXIT
```

5. Add the new lines to the file just before the \$ EXIT line. For example, if 273 is the EXIT command line, you would type the following:

```
* I 273 <Enter> [inserts a new line before line 273 when 273
exists]

DEFINE /SYSTEM CNXRMSE0 SYS$COMMON:[CONNX]CNXRMSE0.COM

<Ctrl> <Z>
* I 273 <Enter> [for this and all subsequent new lines, you repeat
273]
DEFINE /SYSTEM CNXRDBE0 SYS$COMMON:[CONNX]CNXRDBE0.COM
<Ctrl> <Z>
* I 273 <Enter>
DEFINE /SYSTEM CNXDBME0 SYS$COMMON:[CONNX]CNXDBME0.COM
<Ctrl> <Z>
* I 273 <Enter>
DEFINE /SYSTEM CNXSVR SYS$COMMON:[CONNX]CNXSVR.COM
<Ctrl> <Z>
* I 273 <Enter>
DEFINE /SYSTEM CNXDIR SYS$COMMON:[CONNX]
* I 273 <Enter>
@CNXSVR START
<Ctrl> <Z>
* EXIT [to save edit] -or- * QUIT [to not save edit]
$
```

6. Check your work. With the \$ prompt displayed, press the **<up arrow>** key to re-type the following command:

```
edit/edit SYS$MANAGER:SYSTARTUP_VMS.COM
command and then press <Enter>.
```

7. After the * prompt, type `1:END`, which takes you to the end of the file.
8. Observe that the lines you typed are correct. To delete a line, for example, line 275, you would type the following:
`d 275 <Enter>`
9. To change the port number, which by default is set to 6500, include the following before the EXIT command line:

Define/System CNXRUNPORT nnnn

where nnnn is the new port number. If you change the port number on the server, you must also change it on each client machine. See TCP/IP Connection Port in CONNX Registry File Settings for more information on changing the port number on client machines.

10. Delete and retype lines as necessary. When finished, type **EXIT** to save the changes.
11. Disconnect from the session and close your terminal emulator.

Post-Installation Checklist - Sybase data sources

The following table provides a checklist of the post-installation operations you must conduct before the Sybase data source installation is complete. After each operation is complete, place a check mark in the appropriate box. Each operation has a reference to a procedure.

Post-Installation Checklist - Sybase data sources

Check here when complete	Item
	1. Verify that the Data Source Name (DSN) CONNX32 is on the User DSN list of ODBC drivers. Click the Start button, and then point to Settings. In Control Panel, double-click the 32bit ODBC icon. On the User DSN tab, verify that CONNX32 appears as both Name and Driver. If the CONNX32 DSN is not present, it has been erased. This DSN is essential to CONNX operation and must not be erased. To recover the erased CONNX32 DSN, repeat Step 2 - Install CONNX Administrator and CONNX Client .
	2. Using the test data and a utility program supplied with the CONNX installation, establish a connection with the Sybase test data. For a procedure on how to test the connection, see Testing a Database Connection .
	To customize how CONNX is initialized, see CONNX REGISTRY FILE SETTINGS in the CONNX User Reference online help files.

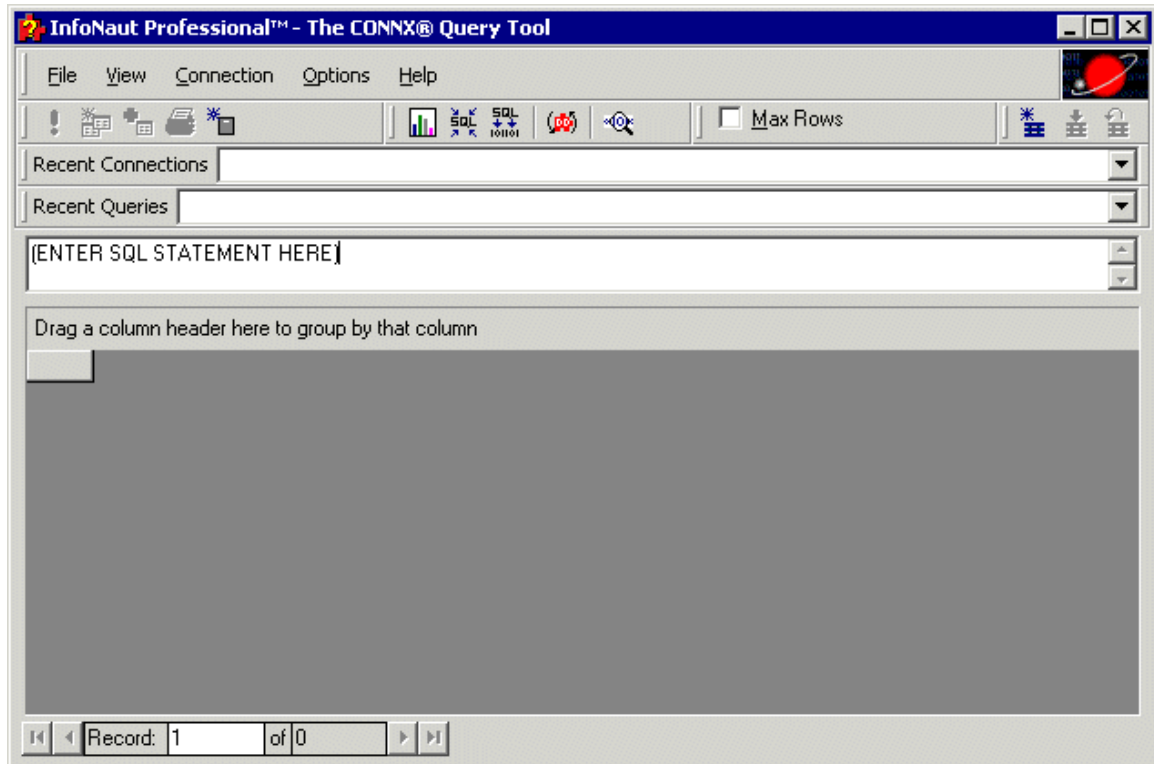
Troubleshooting Post-Installation Procedures

Testing a Database Connection

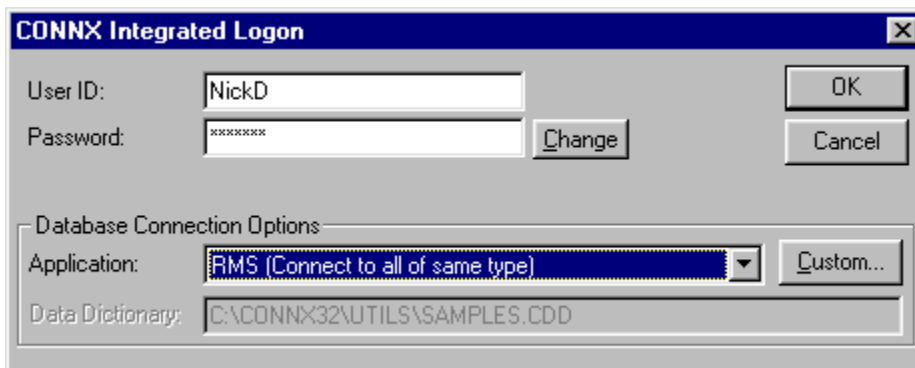
Complete the following procedure to access test data using InfoNaut – The CONNX Query Tool.

To test the database connection using InfoNaut

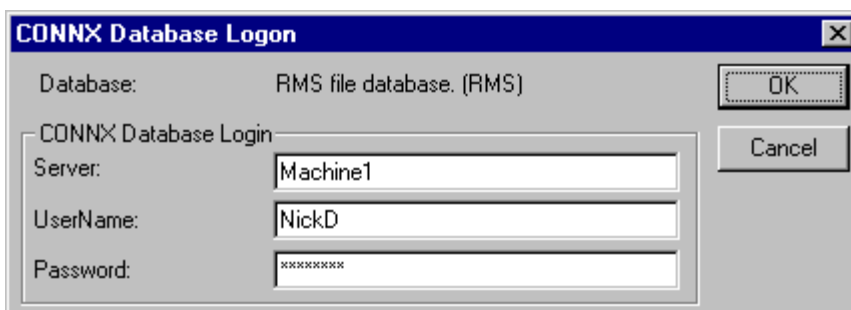
1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **InfoNaut - The CONNX Query Tool**. The InfoNaut window appears.



2. Select **Connections** on the menu bar, and then click **Add New Connection**. The Select Data Source window appears. Click the **Machine Data Source** tab.
3. Select **CONNXxSAMPLES32**, and then click the **OK** button.
4. The CONNX Integrated Logon window appears. Type your server user name and password and then enter an installed CONNX application, for example, **NickD**, **Password**, and **RMS**. Click the **OK** button.



5. The CONNX Integrated Security window appears. Click the **Yes** button to add this user to the Samples CDD.
6. The CONNX Database Logon window appears, listing the selected database. Type the name of your data server, and then click the **OK** button.



7. The **InfoNaut Connection Established** information box appears. Click the **OK** button.

TROUBLESHOOTING

The message "Unable to open connection to server" indicates one of the following:

The user name or password you used is not valid.

Verify the correct user name and password and try again.

The data server was too busy.

Wait a few seconds and try again.

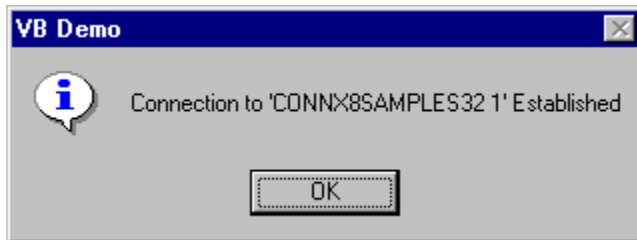
The network is down and connection with the data server is not possible.

Check with your network administrator and try again when network is corrected.

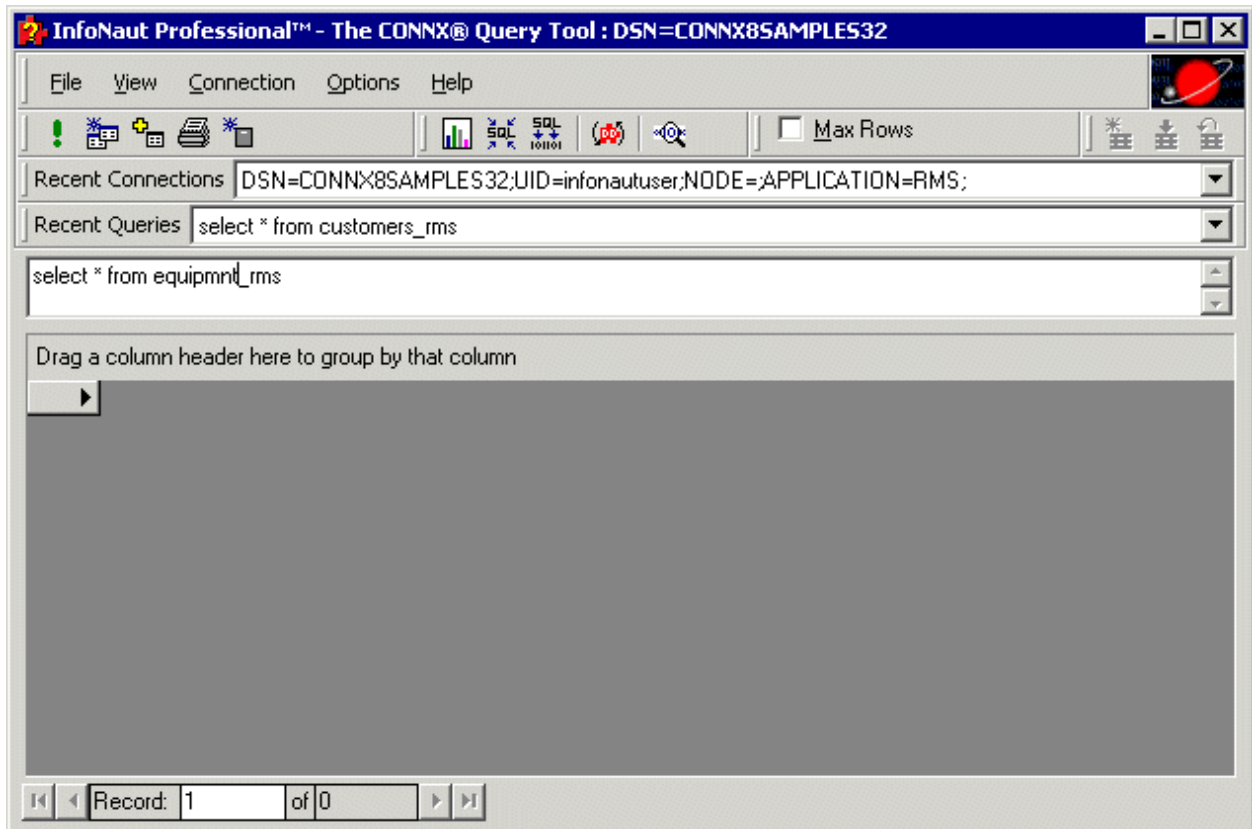
You did not select the correct data source name from the DSN list.

Verify the correct data source name and try again.

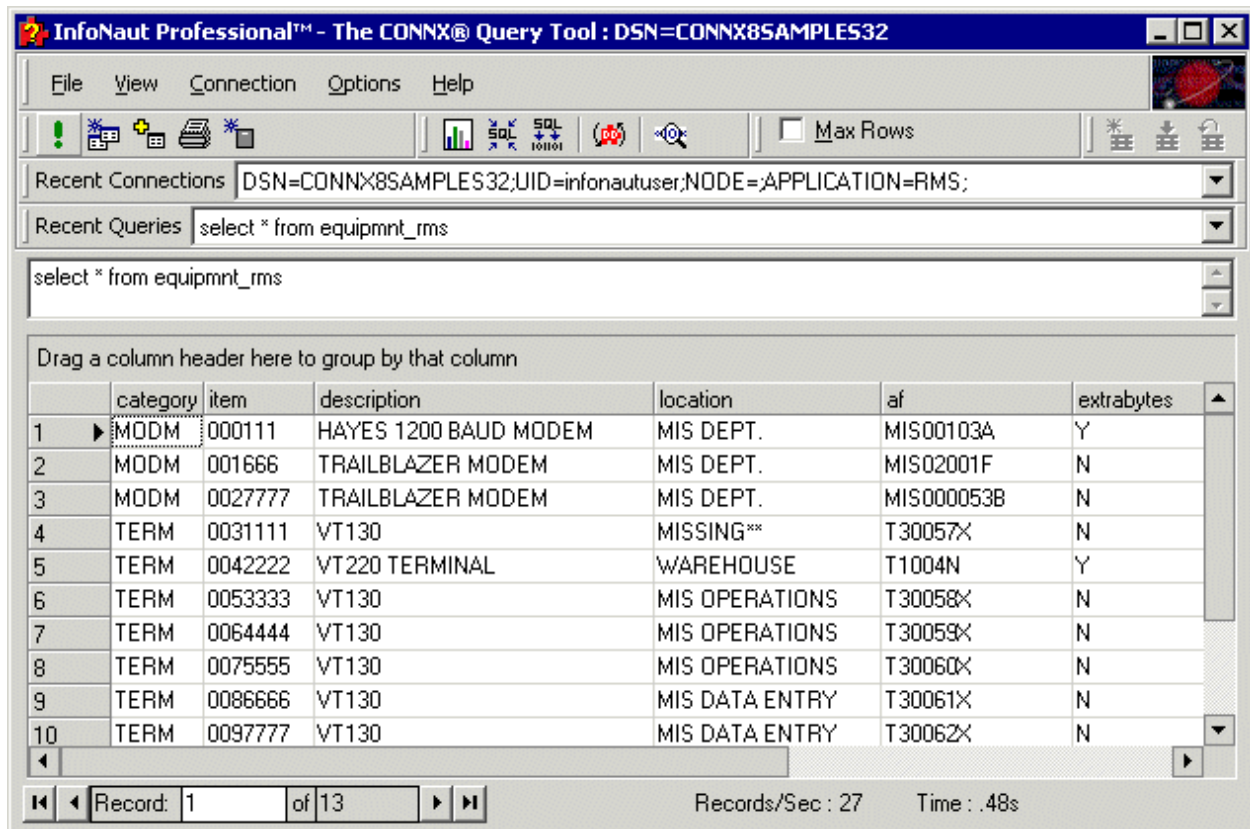
8. The **InfoNaut** dialog box appears. Click the **OK** button.



9. InfoNaut window appears. Type the command **select * from equipmnt_nnn** replacing **nnn** with the database connection you are testing: **equipmnt_rms**, **equipmnt_rdb**, **equipmnt_dbms**, or **equipmnt_dataflex**, and then click the **Execute Query** button.



10. A list of equipment appears as the query result. Verify that your query results are similar to what is shown below.



11. To test another database, return to the second step in this procedure.
12. When finished testing all applicable databases, click the **Disconnect** button for each connection, then select **File** on the menu bar, and click **Exit**.

Reviewing the Minimum OpenVMS Quotas Procedure

From Item 2 in the table under [Post-Installation Checklist - Open VMS Server](#), complete the following procedure to review the minimum quotas for a selected OpenVMS account.

Note: Use a terminal emulator most familiar to you or see [Using Your Telnet Utility - TCP/IP](#).

To review the minimum recommended quotas

1. On your CONNX administrator computer, start your terminal emulator, select the appropriate OpenVMS server, and then type the desired user name and password. The \$ prompt appears.
2. After the \$ prompt, type the following:
SET DEFAULT SYS\$SYSTEM
3. press **<Enter>**, and then type the following:
RUN AUTHORIZE
4. Press **<Enter>**. The prompt changes from \$ to UAF>.
5. After the UAF> prompt, type the following:
show Nick /Page
and then press **<Enter>**, replacing **Nick** with the desired OpenVMS Account user name. The first page of quotas for this user name are displayed.
5. Look for the network access line: "No access restrictions".
Primary days: Mon Tue Wed Thu Fri
Secondary days: Sat Sun
No access restrictions
Expiration: (none)
Pwdminimum: 2
Login Fails: 0
Pwdlifetime: (none)
Pwdchange: (pre-expired)
6. Look for the following quotas:
Maxjobs: 0
Fillm: 200
Bytlm: 82000
Maxacctjobs: 0
Shrfillm: 0
Pybtlm: 0
Maxdetach: 0
BIOlm: 512
JTquota: 4096
Prclm: 8
DIOlm: 512
WSdef: 1024
Prio: 4
ASTlm: 256
Wsquo: 4096
Queprio: 0
TQEIm: 128
WSextent: 16384
CPU: (none)
Enqlm: 2048
Pgflquo: 82000
7. Compare with the recommended minimum requirements:

Maxjobs: 0*
Bytlm: 82000
BIOlm: 512
JTquota: 4096
Prclm: 8
DIOlm: 512
WSdef: 1024
WSquo: 4096
WSextent: 16384
Enqlm: 2048
Pgflquo: 82000

8. Press the **<space bar>** to display the rest of the page, then disconnect from the session and close your terminal emulator.

Note: When Maxjobs is 0, there is no limit to the number of simultaneous users who can access this particular OpenVMS account. When Maxjobs is set to a particular number, then access is limited and some users may be denied access. See your OpenVMS administrator for details.

Note: Adjust the minimum recommended CONNX quotas to meet the requirements of your VMS software.

Minimum Quota Values for RMS, Rdb, and DBMS Databases

When using CONNX for RMS, Oracle Rdb, and Codasyl DBMS, make sure the VMS account has the following minimum quota values:

Maxjobs: 0
Bytlim: 82000
BIOlm: 512
JTquota: 4096
Prclm: 8
DIOlm: 512
WSdef: 1024
WSquo: 4096
WSextent: 16384
Enqlm: 2048
Pgflquo: 82000

Confirm that network access is allowed:

Network Access: No Access Restrictions

Note: When Maxjobs is 0, there is no limit to the number of simultaneous users who can access this particular OpenVMS account. When Maxjobs is set to a particular number, then access is limited and some users may be denied access. See your OpenVMS administrator for details.

Related Topics

 [CONNX OpenVMS Installation Procedure](#)

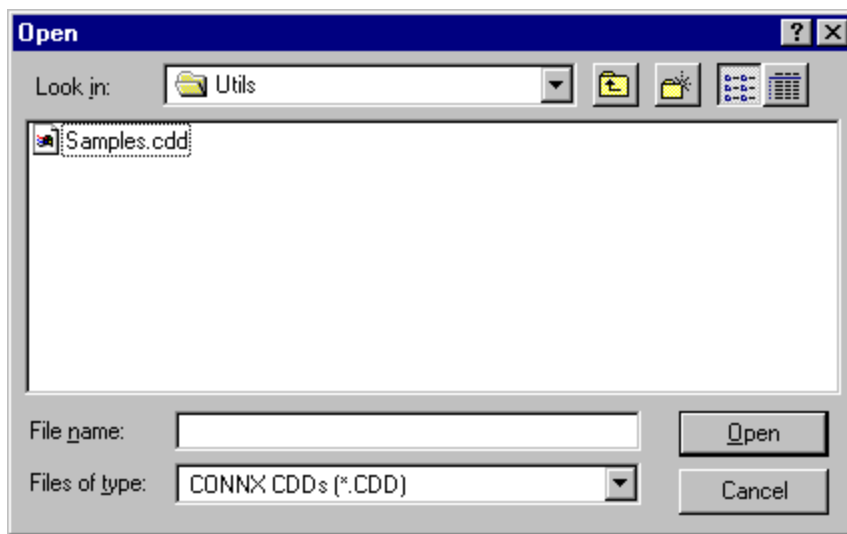
 [Reviewing the Minimum OpenVMS Quotas Procedure](#)

Converting Data Dictionaries

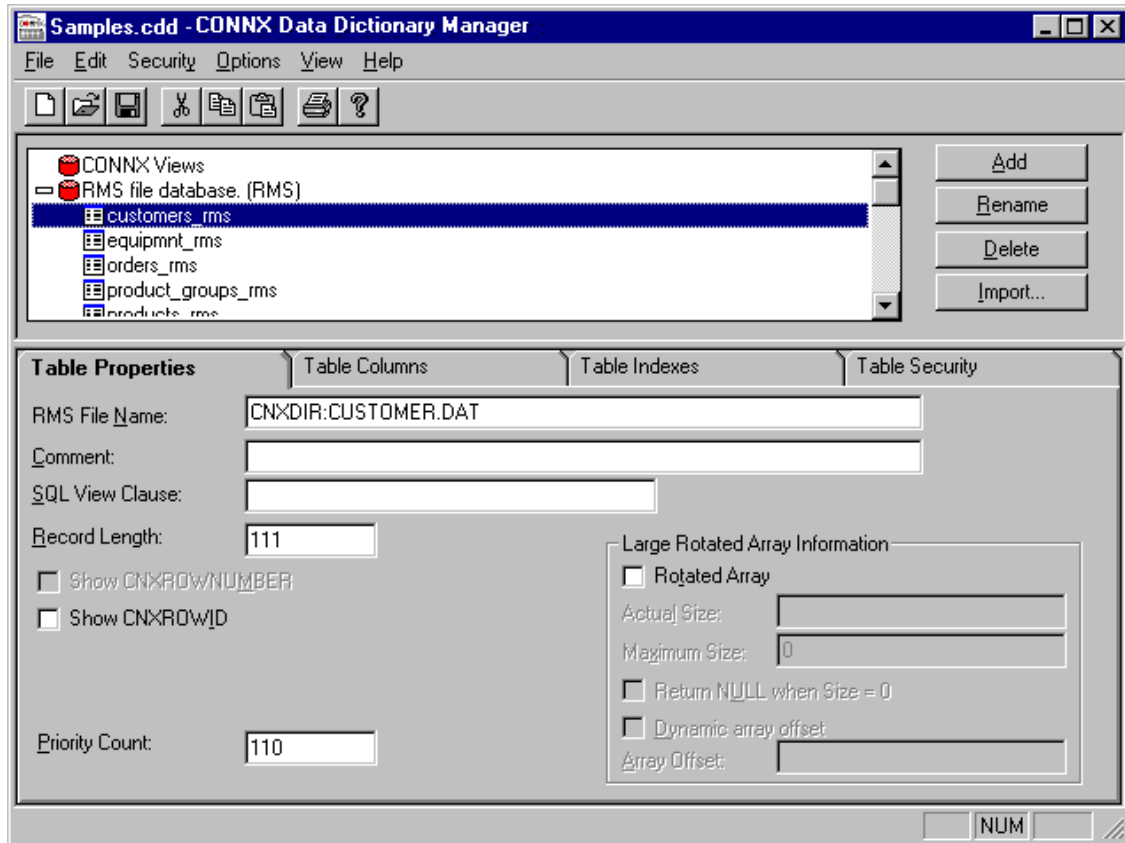
When you upgrade to the latest version of CONNX, you must convert your CONNX Data Dictionary (CDD) files to be compatible with the new version. Complete the following procedure to convert the CDDs.

To convert CONNX Data Dictionaries to the latest version

1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Data Dictionary**. Type your CONNX user name, password, and server in the logon window, and then click **OK**.



2. The **Open** dialog box appears. Select the first of the CDD files to convert, and then click the **Open** button.
3. The selected file opens. Select **File** on the menu bar, and then click **Save**.



The file is automatically converted and updated to the new CDD format. Answer **Yes** to any conversion message.

4. To open the next CDD file to convert, select **File** on the menu bar and then click **Open**. Select the next file to convert. Then select **File** on the menu bar and click **Save**. Use this method to convert each CDD to the new CONNX version.

When all CDD files are converted, select **File** on the menu bar, and then click **Exit** to close the CONNX Data Dictionary.

Important: If you wish to continue using older versions, as well as newer versions, of CONNX, you need to utilize a CDD using the older format. Remember to create backup copies and rename your new CDD files after conversion.

Troubleshooting Connection Problems

When running the CONNX Data Dictionary program from the CONNX administrator, an error message may appear after login information is supplied for the OpenVMS system running CONNX. The two messages that appear may either read:

"Unable to connect. Please make sure your User ID, Password, and Node are correct. Unable to open connection to server."





or

"Unable to open connection to server."

Some common causes and solutions for these connection problems are as follows:

- **Possible problem**
Communication problems exist between the computer and server.
- **Solution**
Test the communications link by conducting the [Pinging the OpenVMS Server - TCP/IP](#) procedure. If you cannot ping successfully, refer the problem to your network administrator.
- **Possible problem**
Error messages appear during OpenVMS server installation.
- **Solution**
Review the log file of your CONNX installation for OpenVMS server by conducting the [Reviewing CONNXSETUP.LOG using FTP -TCP/IP](#) procedure.
- **Possible problem**
Insufficient user account privileges messages appear.
- **Solution**
Verify that a particular user account has the required installation privileges by conducting the [Testing User Account Privileges](#) procedure.
- **Possible problem**
CONNX does not start when the OpenVMS server is booted
- **Solution**
Verify you added the necessary lines to the OpenVMS server SYSTARTUP file, or that the lines were typed correctly by conducting the [Editing the OpenVMS Site-Specific Startup Command](#) procedure.

Related Topics

-  [CONNX OpenVMS Installation Procedure](#)
-  [Testing OpenVMS Server Installation](#)
-  [About the OpenVMS User Account and Password](#)
-  [Testing User Account Privileges](#)

Troubleshooting with RMS, Oracle Rdb, and Codasyl DBMS Databases

- **Possible problem**

The network connection to the OpenVMS system is not working or your OpenVMS login information is incorrect.

- **Solution**

Make sure you can access the OpenVMS system from your computer using the server name, user name, and password you are using with CONNX.

If you are using TCP/IP to make the connection to your OpenVMS system, use a TCP/IP utility such as Telnet or FTP to verify that you can connect to the system.

Use the same server name or address, OpenVMS user name, and password that you are using with CONNX.

If you cannot make a connection from your computer with any of the utilities, contact your network administrator for help in resolving network problems.

- **Possible problem**

The OpenVMS login information stored for your CONNX integrated login account in the CONNX CDD is incorrect.

- **Solution**

Use the Database Account Management option under Security on the File menu to verify that the stored OpenVMS login information for the CONNX integrated login account is correct. Modify the information for your account and save your changes by selecting Save on the File menu.

- **Possible problem**

The CONNX TCP/IP Spawner process is not running on your OpenVMS system.

- **Solution**

The CONNX TCP/IP Spawner process listens for incoming connection requests from computers running the CONNX client software. The process creates CONNX server processes in response to connection requests from the CONNX client software.

The name for the server process is CNXRUN90_MAIN or CNXRUN90_MAIN. The following DCL commands will determine whether the process is running on a selected OpenVMS system:

```
$ SHOW SYSTEM /OUTPUT = TEMP.TXT
$ SEARCH TEMP.TXT CNXRUN
```

If the second command generates the message "%SEARCH-I-NOMATCHES, no strings matched," then the CONNX TCP/IP Spawner process is not running on the system. Start the CONNX TCP/IP Spawner process with the following DCL command:

```
$ @CNXSVR START
```

This command requires that the CONNX logical name CNXSVR be defined as equivalent to the OpenVMS file specification for the CNXSVR.COM file (normally, SYS\$COMMON:{CONNX}CNXSVR.COM).

Related Topics

» [CONNX OpenVMS Installation Procedure](#)

» [Editing the OpenVMS Site-Specific Startup Command](#)

» [Pinging the OpenVMS Server - TCP/IP](#)

Troubleshooting with TCP/IP and OpenVMS

When using TCP/IP, confirm that the CONNX Listener process is running on the VMS server. This can be done by executing the following command:

```
$SHOW SYSTEM
```

Look for the process named CNXRUNAO_MAIN on your OpenVMS server.

If this process does not appear, run **@CNXSVR START** at the \$ prompt, and confirm that this line is in your VMS Startup script.

Make sure the CONNX logicals are defined. This can be done by executing:




```
$SHOW LOG CNX*
```

The following logicals should appear:

```
CNXDIR = SYS$COMMON:[CONNX]
CNXSVR = CNXDIR:CNXSVR.COM
CNXRMSAO = CNXDIR:CNXRMSAO.COM (for RMS users)
CNXDBMAO = CNXDIR:CNXDBMAO.COM (for DBMS users)
CNXRDBAO = CNXDIR:CNXRDBAO.COM (for Rdb users)
```

If these logicals do not appear, CONNX server installation may not have completed successfully and can be restarted.

Related Topics

-  [CONNX OpenVMS Installation Procedure](#)
-  [Editing the OpenVMS Site-Specific Startup Command](#)
-  [Pinging the OpenVMS Server - TCP/IP](#)

Pinging the OpenVMS Server - TCP/IP

Complete the following procedure to ping (verify communications) between your CONNX administrator computer and an OpenVMS server on a TCP/IP network.

To verify the TCP/IP network connection with the OpenVMS server

1. Click **Start** on the taskbar, and then click **Run**. In the **Open** text box, type the following:
`cmd`
and then click the **OK** button. The command window appears with a command prompt.
2. After the command prompt (the path may vary and does not matter), type the following:
`c:\windows>Ping ServerName`

replacing ServerName with the name of your OpenVMS server or the IP address (for example, 123.345.567.0), and then press <Enter>.

The ping results for successful communications are as follows:

```
c:\windows>Ping machine1
Pinging machine1.com [32.333.333.333] with 32 bytes of data:
Reply from 32.333.333.333: bytes=32 time=1ms TTL=255
Reply from 32.333.333.333: bytes=32 time=1ms TTL=255
Reply from 32.333.333.333: bytes=32 time=1ms TTL=255
Reply from 32.333.333.333: bytes=32 time=1ms TTL=255
c:\windows>
```

Refer all network problems to your network administrator for resolution.

3. Close the command window.

Related Topic

 [Reviewing CONNXSETUP.LOG Using FTP - TCP/IP](#)

Reviewing CNXSETUP.LOG Using FTP - TCP/IP

A CNXSETUP.LOG log file is created when you install the OpenVMS server components (Step 4 - Post-Installation Checklist). This log file helps identify possible installation difficulties. To simplify reviewing the log, copy the log file to your CONNX administrator computer desktop using File Transfer Protocol (FTP).

The procedure below uses the FTP client supplied with Windows. Adapt this procedure to suit the FTP client of your choice.

To copy CNXSETUP.LOG using FTP

1. Click **Start** on the taskbar, and then click **Run**. In the **Open** text box, type **ftp** and then click **OK**. The FTP window with the ftp> prompt appears.

Note Depending on the configuration or version of your Web browser, you can also launch directly using the full path name: **c:\windows\ftp.exe**

2. Follow this abbreviated sequence of commands:

1.	ftp> open machine.
2.	Connected to machine1.com
3.	User (machine1): nick
4.	Password
5.	ftp> dir connxsetup.log
6.	CNXSETUP.LOG
7.	ftp> 1cd c:\windows\desktop
8.	ftp> get cnxsetup.log
9.	Transfer completed.
10.	ftp> bye



Description:

1.	Open connection with server.
2.	Connection verified.
3.	Enter the same user name as the one used for the OpenVMS server installation.
4.	Enter the user name and password.
5.	Verify that the CNXSETUP.LOG is present in the [Username] root directory.
6.	Return indicates CNXSETUP.LOG is in the [Username] root directory.
7.	Set the destination folder on the administrator computer to the computer desktop.
8.	Get the CNXSETUP.LOG file and transfer it to the computer desktop.

9.	Return indicates transfer is complete.
10.	End the ftp session and close the ftp window.

3. Locate the **CNXSETUP.LOG** file on your computer desktop. Right-click the file once, and then click **Open With**. Scroll to the desired text editor or word processing program to open the log, and then click the **OK** button.
4. Review, print, and process the **CNXSETUP.LOG** file.

Related Topics

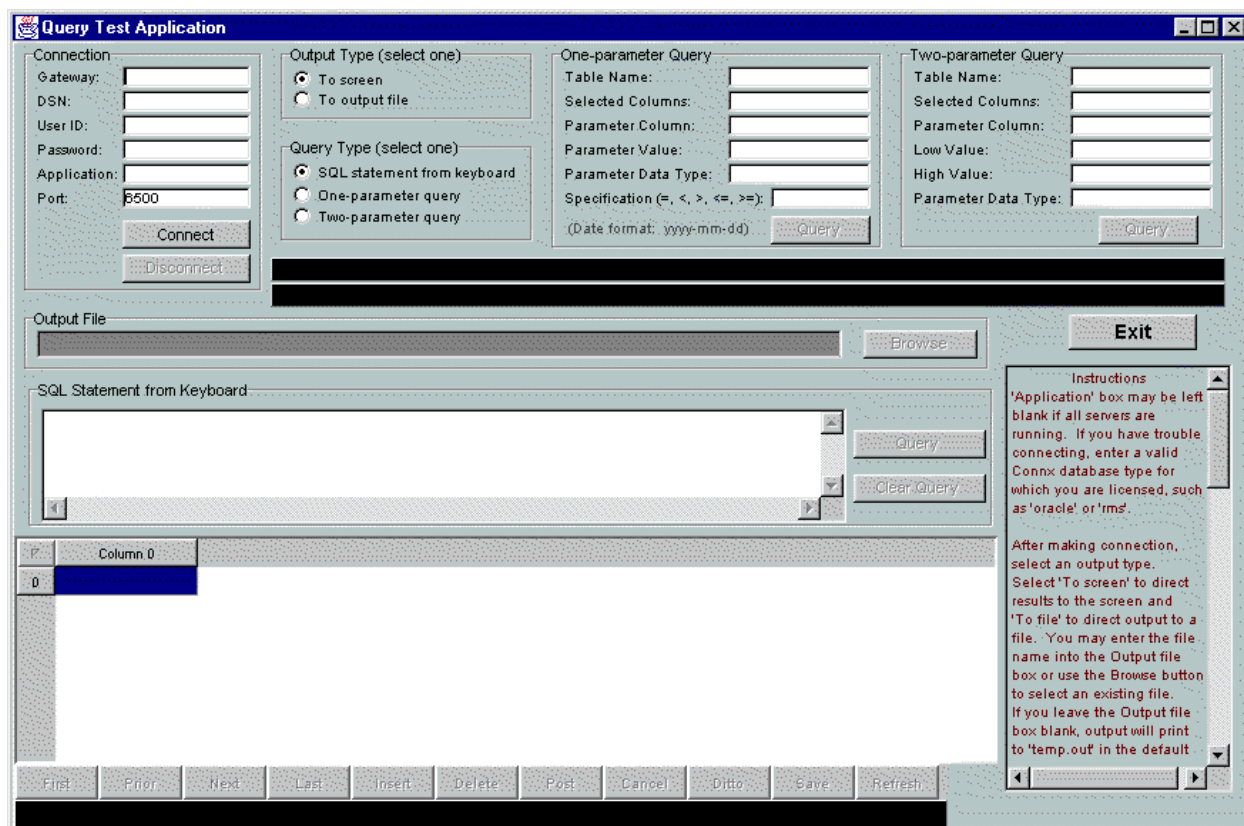
-  [General Information Regarding the Post-Installation Checklist](#)
-  [Pinging the OpenVMS Server - TCP/IP](#)

CONNX JDBC Sample Application

CONNX JDBC Sample Application

The CONNX JDBC Sample Application demonstrates CONNX JDBC in both a Windows and non-Windows environment. The sample application enables the user to formulate one- or two-parameter queries or to enter standard SQL statements directly from the keyboard.

With queries that use parameters, the application demonstrates data type conversion, which enables the user to cast a column of data from its native data type to a more convenient date type. For example, a Web page designer might want to retrieve all data as String, even though the native data types might include Integer, Float, etc.

**Related Topics**

>> [CONNX JDBC General Information](#)

>> [JDBC Resources](#)

>> [Post-Installation Checklist - CONNX JDBC Server](#)

Using the CONNX JDBC Sample Application in a Microsoft Windows Environment

The following procedure describes how to start the CONNX JDBC Sample Application in a Microsoft Windows environment.

To start the CONNX JDBC Sample Application in a Windows environment

1. Click the **Start** button, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX JDBC Sample Application**.
2. The Query Test Application window appears.

Related Topic

[Running the CONNX JDBC Sample Application](#)

Using the CONNX JDBC Sample Application in a non-Windows Environment

The following procedure describes how to start the CONNX JDBC Sample Application in a Microsoft Windows environment.

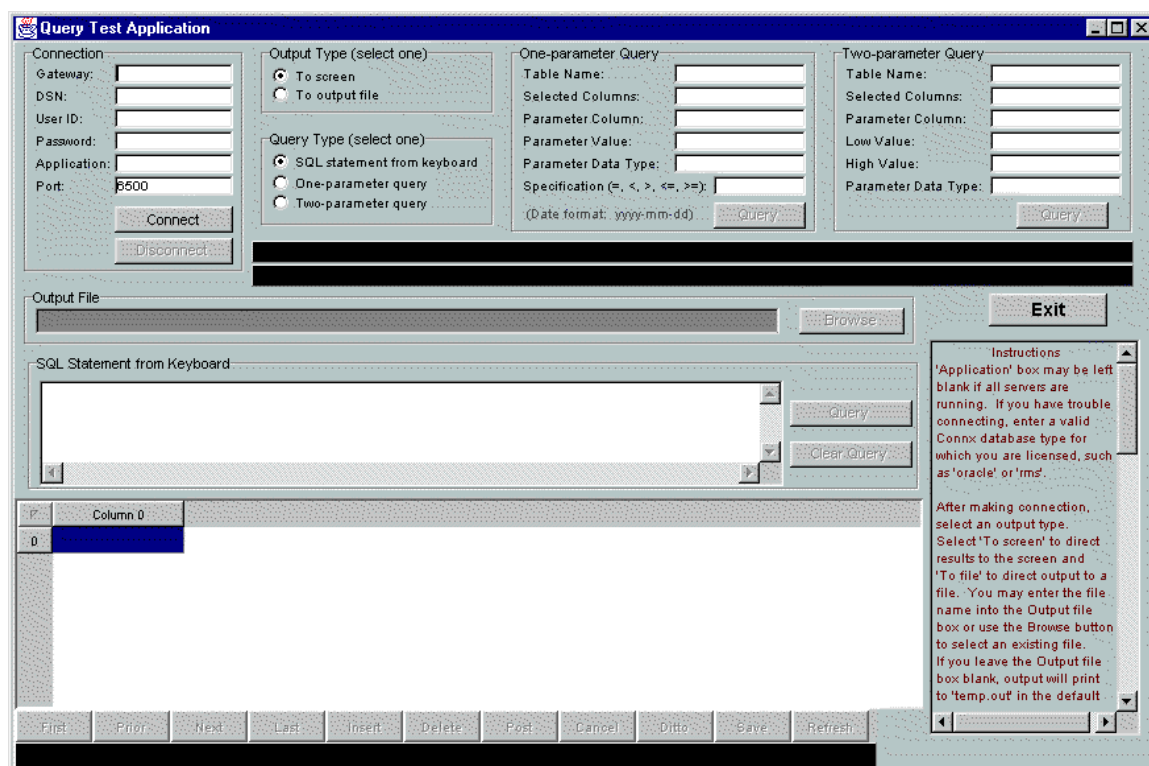
To start the CONNX JDBC Sample Application in a non-Windows environment

1. Verify that the connxjdbcftp.jar file has been copied and unpacked to the non-Windows target. For information on how to transfer the connxjdbcftp.jar file, refer CONNX and JDBC in the CONNX User Reference Guide, available online, on the CD-ROM, and as hard copy.
2. Go to the <installation location>\CONNXJDBC\JAVA\SAMPLES directory. For example, in a UNIX environment, type the following on the command line:

```
cd java/samples
```
3. Press <Enter>, and then type the command

```
sh sapplic_unix
```

The Query Test Application window appears.



4. Follow the instructions on how to log on and run a query in [Running the CONNX JDBC Sample Application](#).

Running the CONNX JDBC Sample Application

The following procedure describes how to connect to a data source, choose an output type, select a query type, and run three different query types.

To connect to the data source

1. Type the following information in the **Connection** group box in the Query Test Application window:

The screenshot shows a 'Connection' dialog box with the following fields and buttons:

- Gateway: [Empty text box]
- DSN: [Empty text box]
- User ID: [Empty text box]
- Password: [Empty text box]
- Application: [Empty text box]
- Port: [7500]
- Buttons: Connect, Disconnect

- **Gateway:** The local server TCP/IP address or host name (for example, 127.0.0.1). To determine the existence of a valid Gateway, open an MS-DOS prompt, and type `ping localhost`.
- **DSN (Data Source Name):** The data source name as registered with the DSNRegistry tool on the CONNX JDBC Server machine. When CONNX is installed, the CDD DSN `connxSamples` is automatically created and registered to enable access to the CONNX8Sample database. The DSNRegistry tool is described in CONNX JDBC Server in the CONNX User Reference Guide, available online, on the CD-ROM, and in hard copy.
- **User ID:** CONNX user name (for example, NickD).
- **Password:** CONNX password (appears as `*****`).
- **Application:** Database connection option (for example, Oracle or RMS). Database connection options are described in CONNX System Requirements, in Step 1, Installation Checklist and System Requirements. The Application text box may be left blank if all servers are running that contain the databases listed in the CONNX Data Dictionary. If any of the servers are down, or if the server running the application specified in the Application text box is down, the connection times out and the following message appears in the SQL Statement from Keyboard text box:

Unable to open database.

```
>> STAT:2 ERR:-12505 SQLSTATE:HYT01 ODBC State = HYT01: Connection
  timeout expired; TCP/IP Error = 10060: The connect request timed
  out.
```

1. **Port:** The port number (default is 7500).
2. Click the **Connect** button in the **Connection** group box. The following message appears in the **SQL Statement from Keyboard** group box:
 Attempting to connect
3. After the connection to the data source is established, the **Disconnect** button in the **Connection** group box and the **Query** and **Clear Query** buttons in the **SQL Statement from Keyboard** group box are activated. The following message appears in the **Miscellaneous Status Bar**:

Connection successful-please select an output type and a query type

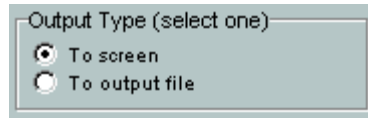
The screenshot shows the 'Query Test Application' window. The 'Connection' section on the left has the following fields: Gateway (127.0.0.1), DSN (connxSamples), User ID (NickD), Password (masked), Application (oracle), and Port (5500). Below these are 'Connect' and 'Disconnect' buttons. The 'Output Type (select one)' section has two radio buttons: 'To screen' (selected) and 'To output file'. The 'Query Type (select one)' section has three radio buttons: 'SQL statement from keyboard' (selected), 'One-parameter query', and 'Two-parameter query'. The 'One-parameter Query' section includes fields for Table Name, Selected Columns, Parameter Column, Parameter Value, Parameter Data Type, and Specification (=, <, >, <=, >=). The 'Two-parameter Query' section includes fields for Table Name, Selected Columns, Parameter Column, Low Value, High Value, and Parameter Data Type. A status bar at the top displays the message 'Connection successful-please select output type and query type'. Below the connection settings is an 'Output File' section with a text box and a 'Browse' button. The 'SQL Statement from Keyboard' section has a large text area with 'Query' and 'Clear Query' buttons. At the bottom is a table with one row and one column, labeled 'Column 0' and '0'. A right-hand pane contains 'Instructions' and a 'Query' button. The bottom of the window features a series of navigation buttons: First, Prior, Next, Last, Insert, Delete, Post, Cancel, Ditto, Save, and Refresh.

Related Topic

[Running the CONNX JDBC Sample Applet](#)

To choose an Output Type

- Select the **To screen** or the **To File** option in the **Output Type** group box.

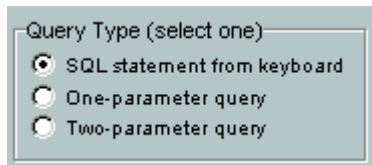


The image shows a small rectangular window titled "Output Type (select one)". Inside the window, there are two radio button options. The first option is "To screen", which has a filled-in radio button, indicating it is the selected option. The second option is "To output file", which has an empty radio button.

- The **To screen** option directs output to the computer screen (the default). If you select the **To screen** option, the result set appears in the Data Grid and the Navigator bar becomes active.
- The **To output file** option directs output to an output file. If you select the **To output file** option, the Output File group box is activated. Click the **Browse** button to locate an existing file, or type a file name in the **Output File** text box. The result set of each query is appended to an existing file. If no file is specified, the results of the most recent query overwrite the tmp.out file, located in the \CONNXJDBC\Java\Samples subdirectory where CONNX was originally installed (usually C:\CONNX32).

To choose a Query Type

1. Select the **SQL Statement from Keyboard** (default) option if you want to type an SQL statement directly from the keyboard.



Example:

```
Select * from customers_rms where customerid like 'A%' and  
customerstate = 'WA'
```

2. The **Query** and **Clear Query** buttons are activated in the **SQL Statement from Keyboard** group box. Type an SQL statement into the **SQL Statement from Keyboard** text box. If you end the SQL Statement with a semicolon (;), a syntax error appears.
3. Click the **Query** button to execute the query.
4. Click the **Clear Query** button to clear the **SQL Statement from Keyboard** text box.

To create a one-parameter query

1. Select the **One-parameter Query** option in the **Query Type** group box, and then enter the query information in the **One-parameter Query** group box.

Example:

Select * from orders_rms where productid > 22

- **Table Name:** The name of the table to be queried (orders_rms in the example).
 - **Selected Columns:** The names of the columns in the result set, separated by commas. Enter an asterisk (*) to retrieve all columns from the table.
 - **Parameter Column:** The name of the column that contains the parameter (productid in the example).
 - **Parameter Value:** The parameter that limits the data set (22 in the example).
 - **Parameter Data Type (optional):** The desired data type for the retrieved parameter column (may be different from its native data type). If a Parameter Data Type is not selected, the returned data defaults to String.
2. Click the **Query** button in the **One-parameter Query** group box. The result set from the query appears in the **Data Grid**.

To create a two-parameter query

1. Select the Two-parameter Query option in the Query Type group box.

Example:

```
select * from customers_rms where customerid >= A and customerid <= F
```

2. Enter the query information in the Two-parameter Query group box.
 3. **Table name:** The name of the table to be queried (customers_rms in the example).
 4. **Selected columns:** The names of the columns in the answer set, separated by commas. Enter an asterisk (*) to retrieve all columns from the table.
 5. **Parameter column:** The name of the column that contains the parameter (customerid in the example).
 6. **Low Value:** The low parameter value that sets the lower bound of data (A in the example).
 7. **High Value:** The high parameter value that sets the upper bound of data (F in the example).
 8. **Parameter Data Type (optional):** The data type for the retrieved parameter column (may be different from its native data type). If a Parameter Data Type is not selected, the returned data defaults to String.
3. Click the **Query** button in the **Two-parameter Query** group box.

The image shows a dialog box titled "Two-parameter Query". It contains six input fields with labels: "Table Name:", "Selected Columns:", "Parameter Column:", "Low Value:", "High Value:", and "Parameter Data Type:". Each label is followed by a text input box. At the bottom right of the dialog box is a button labeled "Query".

4. The result set from the query appears in the **Data Grid**.

Navigator Bar

Use the Navigator bar to examine or modify the data set. The data set represents a live link to the original data. Important: Any saved changes to the data set are saved to the original data source if the user has update privileges.



The Navigator bar contains the following keys:

- **First:** Moves the cursor to the first row of the data set.
- **Prior:** Moves the cursor to the previous row of the data set.
- **Next:** Moves the cursor to the next row of the data set. If focus is on the last row when the button is clicked, a new (blank) row opens.
- **Last:** Moves the cursor to the last row of the data set.
- **Insert:** Opens a new (blank) row to allow data entry (must be Saved to be made permanent). Produces an error if user does not have update privileges.
- **Delete:** Deletes the selected row (must be Saved to be made permanent). Produces an error if user does not have update privileges.
- **Post:** If selected row has been edited, saves changes to the selected row only. Important: no changes are permanently saved back to the data source unless the Save button is used and the user has update privileges.
- **Cancel:** Cancels modifications made to the selected row of data. The data is restored to the way it was before the edit began or to the state it was since the last Save.
- **Ditto:** Copies the data from the row above to the selected row.
- **Save:** Saves all changes made since the last Save to the data source. This is a permanent change that affects the original data source. Produces an error if user does not have update privileges.
- **Refresh:** Refreshes the data set from the original data source.

CONNX JDBC Sample Applet

Running the CONNX JDBC Sample Applet

A sample Applet is provided to test CONNX JDBC installation. For this example, CONNXSamples was used as a Data Source Name. To test other data sources, it is necessary to use the DSNRegistry tool to register the data sources. There are two ways to test the applet as follows:

Option 1 (Running the CONNX JDBC Sample Applet from an AppletViewer):

This option requires an AppletViewer application. AppletViewer is a Java SDK application that allows applets to be tested without a Web browser. It may be necessary to install a Java SDK to obtain the AppletViewer application. A Java SDK has more utilities than the JRE (Java Runtime Environment) that is installed with CONNX JDBC. Java SDKs are available through download for systems running JDBC. See JDBC Resources, for a list of Web sites that offer JDK download options.

If you want to test the CONNX JDBC applet from a Windows machine, follow this procedure:

- Click **Start**, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX JDBC Sample Applet**.

If you want to test the CONNX JDBC applet from a non-Windows machine, follow this procedure:

1. Verify that the connxjdbcftp.jar file has been copied and unpacked to the non-Windows target. For information on how to transfer the connxjdbcftp.jar file, refer to [Moving a Client to a non-Windows Platform by Transferring the .jar File](#).
2. Go to the C:\<installation location>\CONNXJDBC\JAVA\SAMPLES directory. For example, for UNIX, type cd java/samples on the command line.
3. Press <Enter>, and then type the following command:
`appletviewer connxSampleApplet.html`

to run the sample applet. The Applet Frame window appears. Follow the instructions described in Option 1 above.

Option 2 (Testing from a Web Server):

Follow this procedure to test the sample applet from a Web server.

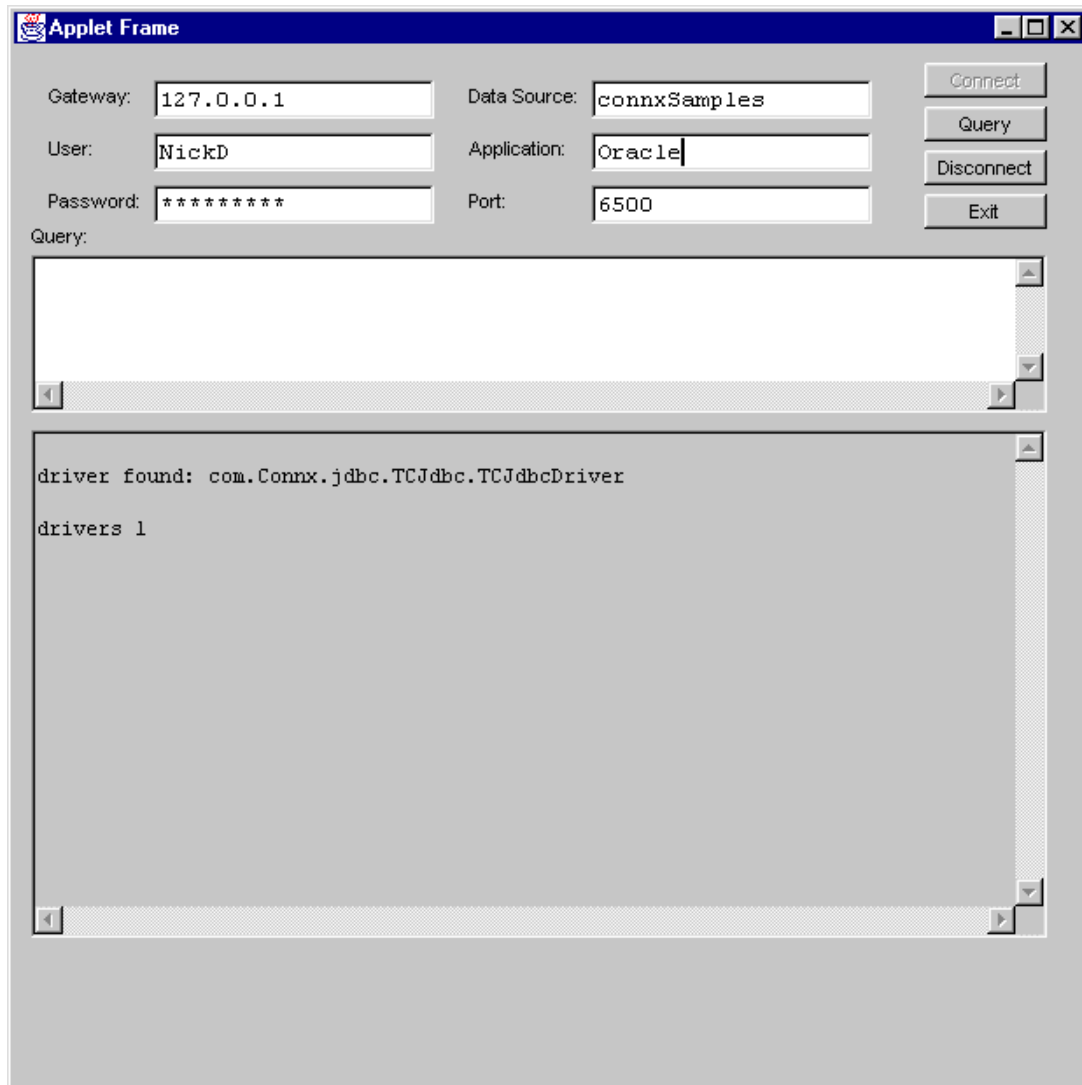
1. Copy the ConnxSampleApplet.html, ConnxSampleApplet.jar, and ConnxJdbc.jar files to a directory that can be accessed by your Web server, for example, C:\inetpub\wwwroot\jdbctest. If the Web server root file is C:\inetpub\wwwroot, client can access the html page by going to http://<website ip address> or <website name>/jdbctest/ConnxSampleApplet.html.
2. Test the installation by using a different client machine to access the ConnxSampleApplet.html file. The client machine does not have to have CONNX JDBC. The ConnxJdbc.jar file is downloaded when the .html file is accessed. If it is a non-Windows Web server, for example, Linux, the CONNX JDBC Router must be installed on the Web server machine.
3. For more information about the CONNX JDBC Router, see the CONNX JDBC Server section in the *CONNX User Reference Guide*, available online and on the CONNX CD-ROM.

To run the CONNX JDBC Applet

1. Click **Start**, and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX JDBC Sample Applet**. The Applet Frame window appears.

2. Type the server TCP/IP address or host name, for example, localhost, in the **Gateway** text box.
3. Type a CONNX user name in the **User** text box.
4. Type a CONNX password in the **Password** text box.
5. Type a CONNX Data Source Name registered in DSNRegistry, for example, ConnxSamples, in the **Data Source** text box.
6. If you are using a CONNX application, type the name in the **Application** text box.
7. Click the **Connect** button to connect.

A message appears in the upper pane confirming connection.



8. Type a query in the **Query** text box, for example,
Select * from customers_oracle.

- Click the **Query** button to run the query.

Applet Frame

Gateway: 127.0.0.1 Data Source: connxSamples Connect

User: NickD Application: oracle Query

Password: ***** Port: 6500 Disconnect

Exit

Query:

Select * from customers_oracle

SAWYH	Sawyer Hill General	234 Samuel Pl.	Ithaca	NY
SEVES	Seven Seas Imports	90 Wadhurst Rd.	London	
SILVS	Silver Screen Food G	12 Meikeljohn Ln.	Helena	MT
SPECD	Special Delights	14 Criswall St.	London	
SUGAA	Sugar and Spice	413 Vestibule Rd.	London	
TOMMI	Tommy's Imported Spe	348 - 33rd St.	Denver	CO
TOPOF	Top of the Food Chai	488 Sacajawea Ct.	Santa Fe	NM
TRADM	Trade More	106 Cravern Rd.	London	
VALUF	ValuMax Food Stores	P.O. Box 10029	Austin	TX
VILLF	Village Food Boutiqu	2782 - 92nd Ave.	Los Angeles	CA
VINEA	Vine and Barrel, Inc	12405 Aurora Ave. N.	Seattle	WA
WALNG	Walnut Grove Grocery	33 Upper Arctic Dr.	Buffalo	NY
WANDW	Wanda's Wine and Che	998 Kirk Rd.	Bellingham	WA
WELLT	Wellington Trading	16 Newcomen Rd.	Sevenoaks	Ker
WHITC	White Clover Markets	1029 - 12th Ave. S.	Seattle	WA
WITAE	Witanowski's East-We	7775 S.W. Clinton Av	Lewiston	ID
Total Rows Processed = 64				

- Click the **Disconnect** button to disconnect the applet.

Step 10 - License Registration

General Information Regarding License Activation

The license activation procedure is required for the following:

- When you first install the CONNX product.
- When you are adding additional users to an existing CONNX installation, for example, upgrading from 25 to 50 users.
- When you are upgrading your existing CONNX installation by adding new features, for example, an RMS database.

If you are re-installing or upgrading CONNX from a version that already has activated licenses and your license has not changed with this CONNX installation, license activation is not required.

If you are upgrading CONNX from one major version to another and the new version does not yet have activated licenses, you will need to activate new licenses for the new version. This applies to major version upgrades only. i.e. when upgrading from version 11.5 to version 12.

The license activation procedure requires the license files shipped to you when you purchased CONNX, or update license codes that have been emailed to you. You can request to have the license update code sent to you by fax, e-mail, letter, or you can request it by telephone.

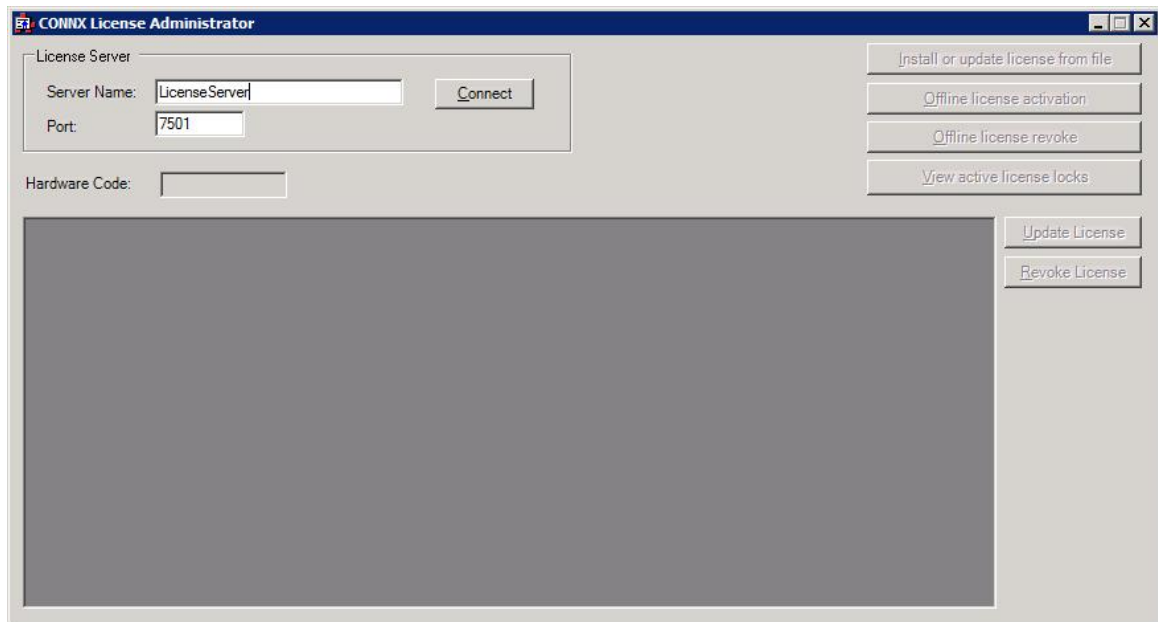
License Activation

Complete the following procedure to install or update licenses, and activate them.

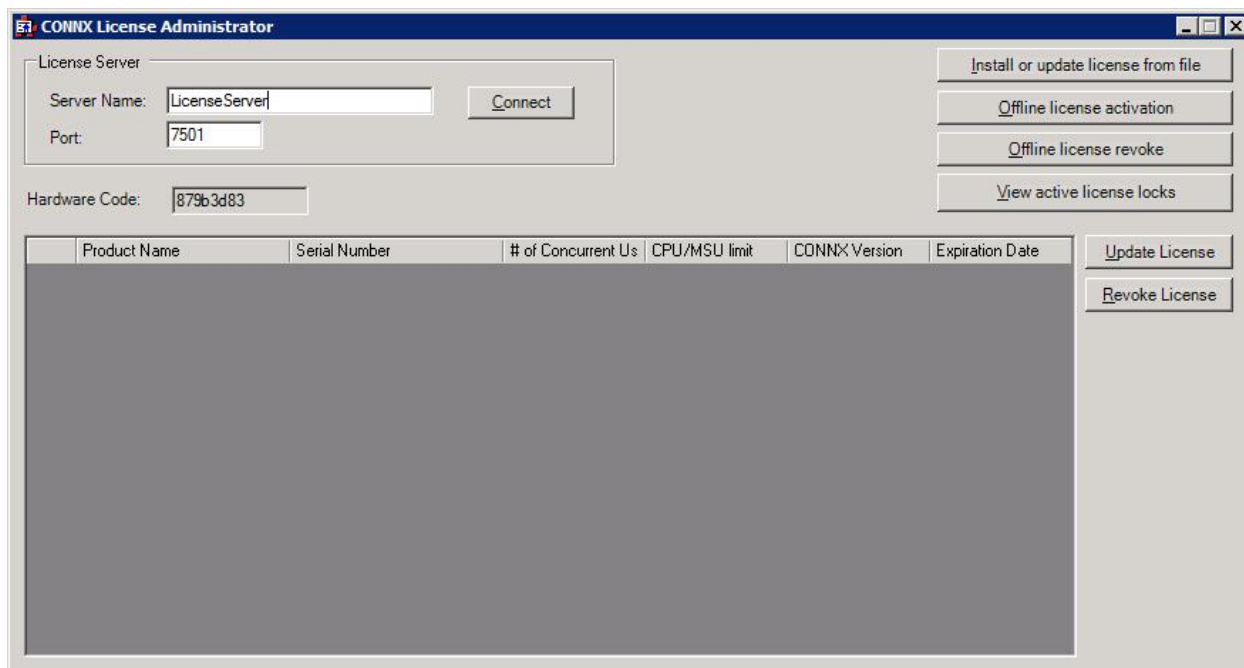
Important: If you uninstall CONNX using the Add or Remove Programs applet in the Windows Control Panel, be sure you copy any .cdd files that you may have created and stored in the CONNX32 folder structure to another location on your hard drive.

1. On a CONNX client computer on which the CONNX Administrator component is installed, click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX License Administrator**. The **CONNX License Administrator** window appears.

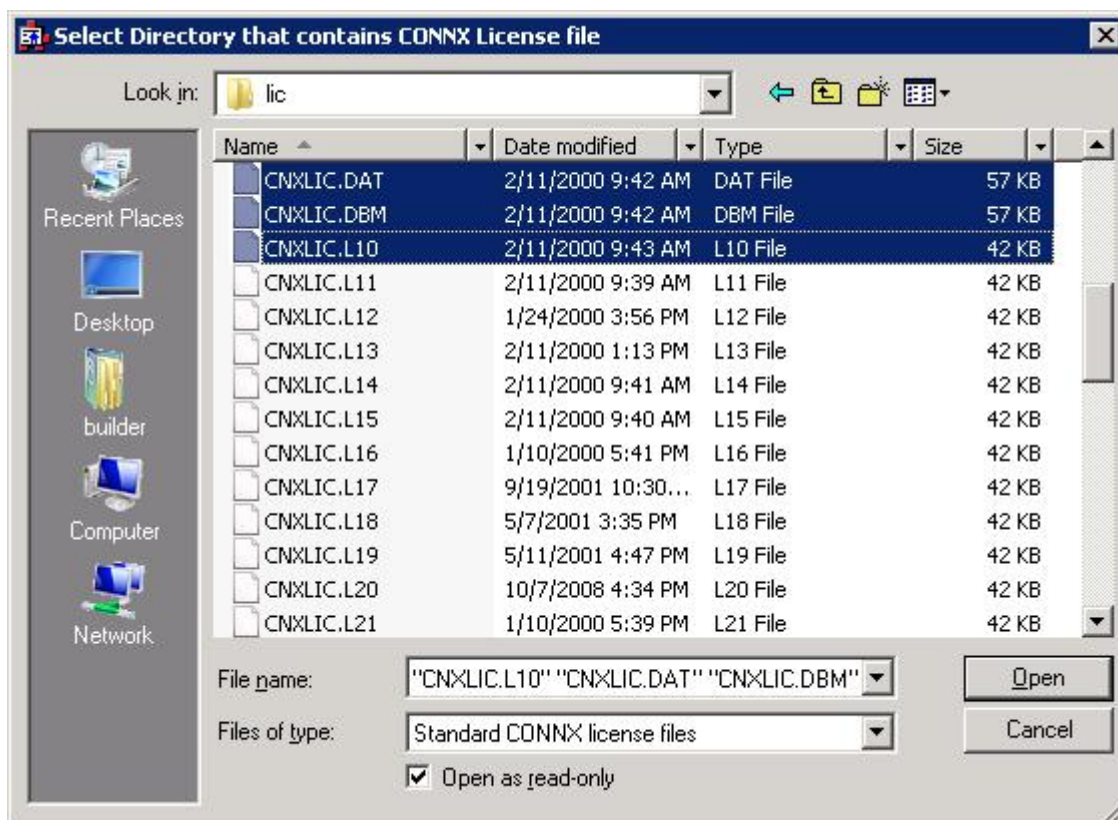
The server name and port should be pre-populated with the correct CONNX license server, based on selections during installation. If the server name and port are not correct, enter the proper information. Press the **Connect** Button.



2. Once connected to a license server, the **Install or Update license from file** button, **Offline license activation** button, **View active license locks**, **Update license**, and **Revoke License** button become enabled. Additionally, the **hardware code** of the license server (only required for off-line license activation) is displayed. If the computer running the license administrator has internet access, then proceed with online license activation by selecting either the **Install or Update license from file** button, or the **Update license** button. .



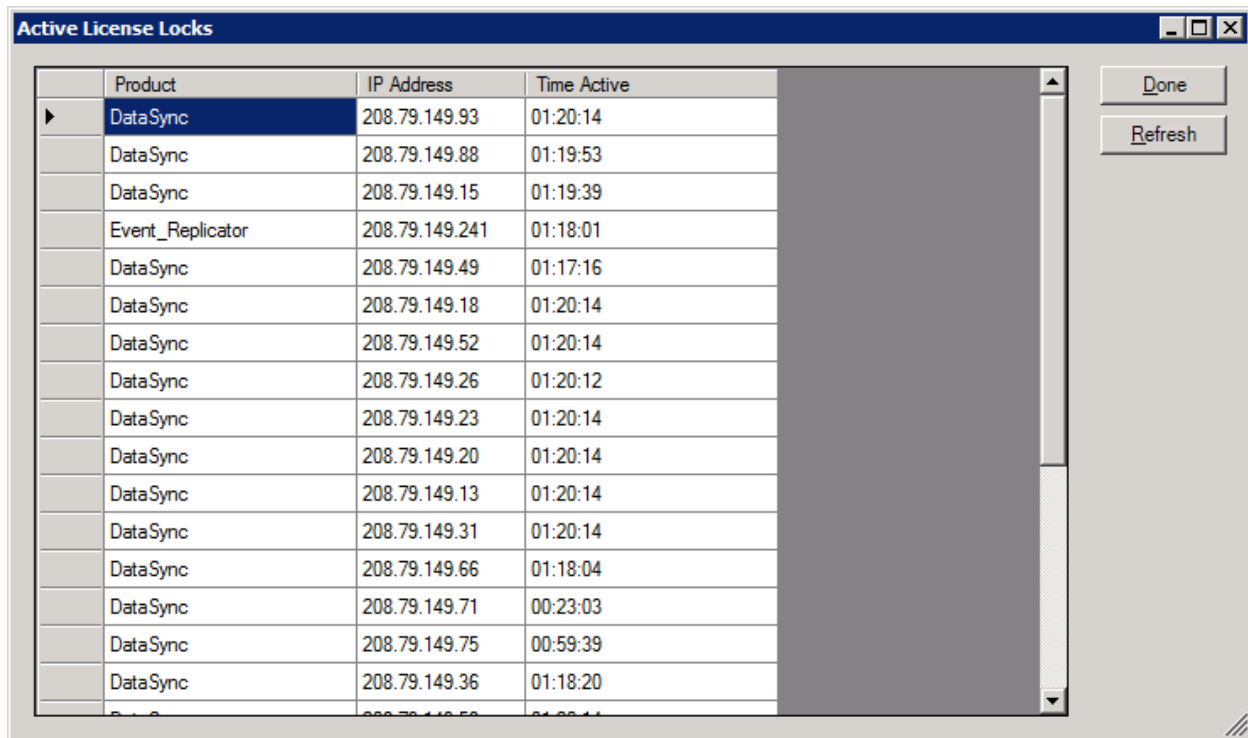
3. If you have one or more license files that need to be installed and activated, select the **Install or Update license from file** button.



4. Select the appropriate license files for installation and activation. If you have a license diskette with all license files on it, you can sort by date and only select the valid license files purchased. Multiple

files can be selected by holding down the CTRL or SHIFT key. Press **Open** once the appropriate files are selected.

5. After several seconds, the **License Installed/Updated Successfully** window appears, confirming valid registration and activation.
6. **Revoking a License:** When a license is activated, it is tied to specific server the license server is running on and may not be activated on any other license servers unless it is first revoked from this license server. To revoke a license, select the license you wish to revoke and press the **Revoke License** button. This will free the license to be activated on a different license server.
7. To list all licenses currently in use, in the **CONNX License Administrator** window, click **View active license locks**.

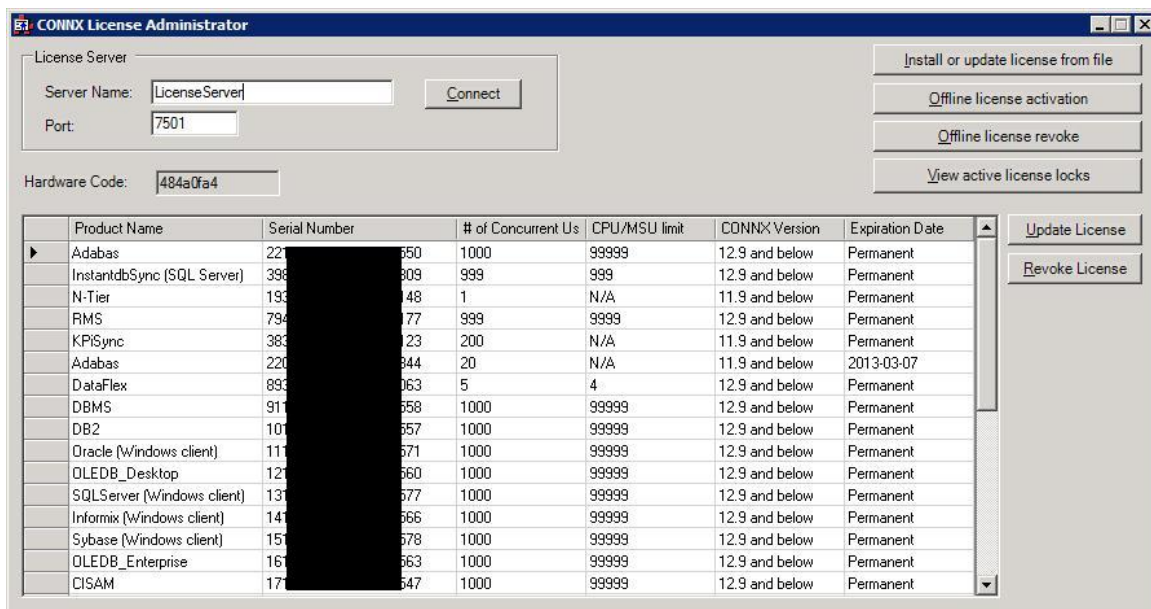


Product	IP Address	Time Active
DataSync	208.79.149.93	01:20:14
DataSync	208.79.149.88	01:19:53
DataSync	208.79.149.15	01:19:39
Event_Replicator	208.79.149.241	01:18:01
DataSync	208.79.149.49	01:17:16
DataSync	208.79.149.18	01:20:14
DataSync	208.79.149.52	01:20:14
DataSync	208.79.149.26	01:20:12
DataSync	208.79.149.23	01:20:14
DataSync	208.79.149.20	01:20:14
DataSync	208.79.149.13	01:20:14
DataSync	208.79.149.31	01:20:14
DataSync	208.79.149.66	01:18:04
DataSync	208.79.149.71	00:23:03
DataSync	208.79.149.75	00:59:39
DataSync	208.79.149.36	01:18:20

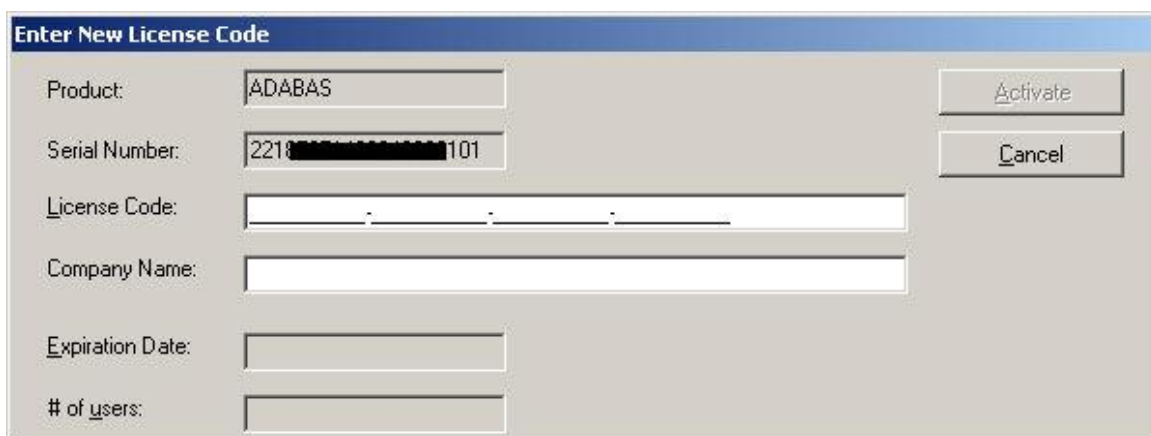
8. Close the **CONNX License Administrator** window, or conduct any additional license upgrade procedures, as required.

Updating an existing license

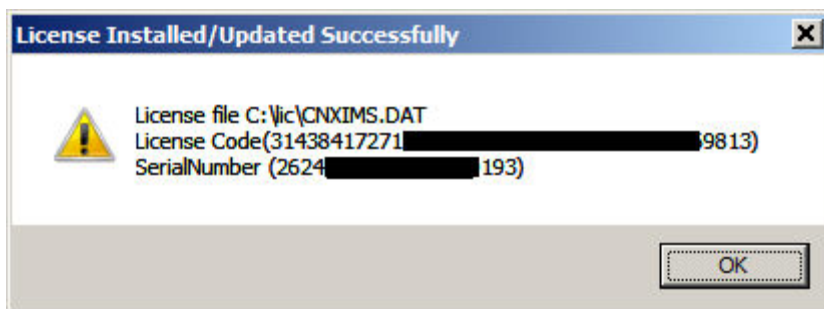
1. On a CONNX client computer on which the CONNX Administrator component is installed, click **Start**, point to **All Programs**, point to **CONNX Solutions**, point to **CONNX**, and then click **CONNX License Administrator**. The **CONNX License Administrator** window appears. The server name and port should be pre-populated with the correct CONNX license server, based on selections during installation. If the server name and port are not correct, enter the proper information. Press **Connect**.



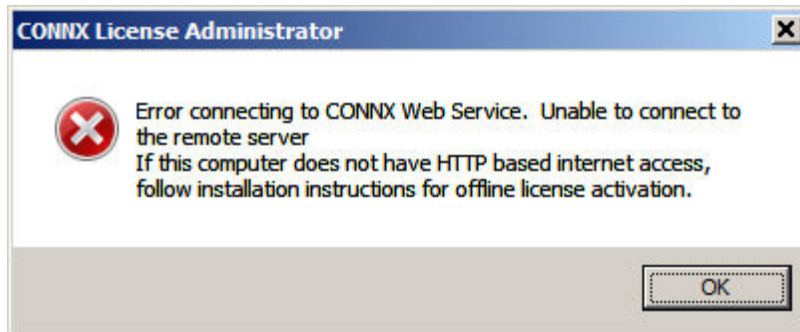
- Once connected to a license server, select a license from the list, then press the **Update license** button.



- Supply the company name and the license code from the license certificate faxed or emailed as part of the CONNX package. Press Activate.
- After several seconds, the **License Installed/Updated Successfully** window appears, confirming valid registration and activation.



- If you get the following message instead at this point, the computer that is running the license administrator may not have internet access, or a corporate firewall may be preventing the activation to occur. In this situation, licenses must be activated using the [Offline License Activation](#) procedure.



6. To list all licenses currently in use, in the **CONNX License Administrator** window, click **View active license locks**.

Product	IP Address	Time Active
DataSync	208.79.149.93	01:20:14
DataSync	208.79.149.88	01:19:53
DataSync	208.79.149.15	01:19:39
Event_Replicator	208.79.149.241	01:18:01
DataSync	208.79.149.49	01:17:16
DataSync	208.79.149.18	01:20:14
DataSync	208.79.149.52	01:20:14
DataSync	208.79.149.26	01:20:12
DataSync	208.79.149.23	01:20:14
DataSync	208.79.149.20	01:20:14
DataSync	208.79.149.13	01:20:14
DataSync	208.79.149.31	01:20:14
DataSync	208.79.149.66	01:18:04
DataSync	208.79.149.71	00:23:03
DataSync	208.79.149.75	00:59:39
DataSync	208.79.149.36	01:18:20

7. Close the **CONNX License Administrator** window, or conduct any additional license upgrade procedures, as required.

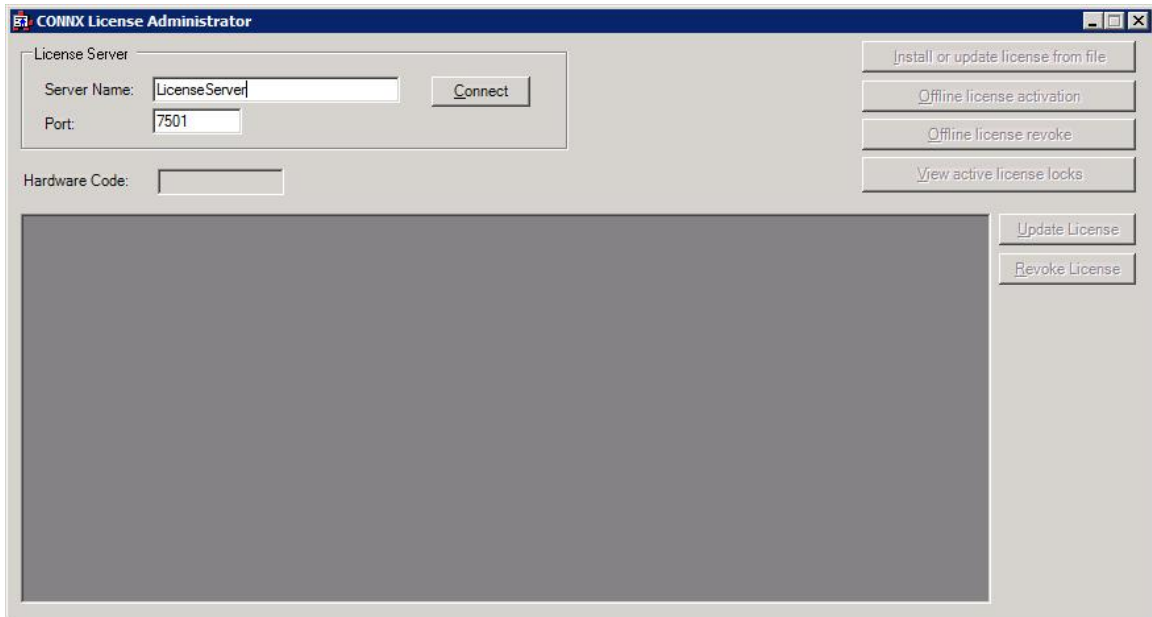
Notes about License Servers and Licenses

- Licenses have an embedded version – meaning that a CONNX 12 client will require a CONNX 12 (or higher) license. A CONNX license will work with a lower version client. For example a CONNX 12 license will work with a CONNX 11.x or CONNX 10.x client. The version of the CONNX license needed for a given version of the client is based on the major version number. For example, a CONNX 12 license will work with any minor version of CONNX 12 (i.e. CONNX 12.x, etc.)
- CONNX requires an N-Tier license when used on a “server” operating system. Server operating systems include all Unix/Linux platforms.

- CONNX checks the CPU count of the data server against the CPU count generated in the license. On Unix/Linux, Windows and VMS platforms, the CPU count is the number of cores the processor or processors have. On an IBM Mainframe platform, the MSU count is used for licensing purposes. In the License Administrator, the column “CPU/MSU limit” specifies the number of CPUs or MSUs the license is valid for.
- A license may be activated only once and is associated with the license server it was activated on.
- The License Administrator has a “revoke” feature which will remove a license from a server, this will allow a license to be moved from one server to another.

Offline License Activation

1. If the computer running the CONNX License Administrator does not have internet access, licenses must be activated in "offline" mode, using the **Offline license activation** button.
2. On a CONNX client computer on which the CONNX Administrator component is installed, click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX License Administrator**. The **CONNX License Administrator** window appears. The server name and port should be pre-populated with the correct CONNX license server, based on selections during installation. If the server name and port are not correct, enter the proper information. Press **Connect**.



3. Once connected to a license server, the **Install or Update license from file** button, **Offline license activation** button, **View active license locks**, and **Update license** button become enabled. Additionally, the **hardware code** of the license server (only required for off-line license activation) is displayed. If the computer running the license administrator has internet access, then proceed with online license activation by selecting either the **Install or Update license from**

file button, or the **Update license** button.

CONNX License Administrator

License Server

Server Name:

Port:

Hardware Code:

Product Name	Serial Number	# of Concurrent Us	CPU/MSU limit	CONNX Version	Expiration Date

4. Select the **Offline license activation** button.

Offline License Code Activation

Serial Number:

Hardware Code:

Authentication Code:

Activation Code:

Company Name:

Expiration Date:

of users:

Please to go <http://www.connx.com/activate> and follow the instructions.

5. Supply the serial number. This will populate the Authentication code. Write down this code, you will need it in the next step.
Important: Keep this window open. Do not close it until you are able to fill in all of the fields.

6. Locate a computer that does have internet access, and go to <http://www.connx.com/activation.asp>.

CONNX Offline License Activation

Offline Activation Instructions:

- 1) Run the license administrator, select offline activation.
- 2) Type or paste in the serial number of the license into the license administrator. This will populate the Authentication code required for the next step.
- 3) Copy or type the Authentication code from the license administrator into this web page.
- 4) Copy or type the serial number, company name, and license code into this web page.
- 5) Press Generate Activation.
- 6) Copy or type the Activation code from this web page into the license administrator.
- 7) Copy or type the company name, license code of the license into the license administrator.
- 8) Press Activate in the license administrator.

NOTE: For old builds of CONNX prior to CONNX 12 build 14267, click [here](#).

Serial Number:

Hardware Code:

Company Name:

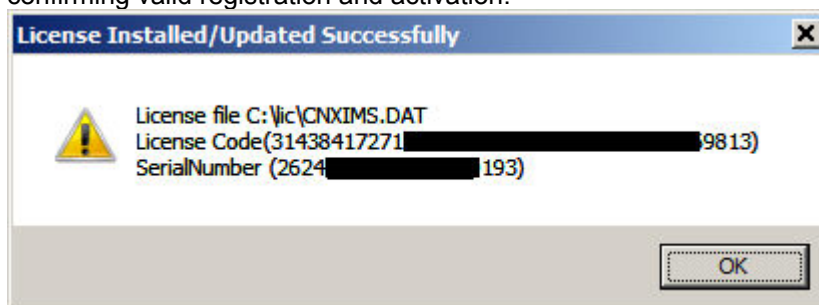
License Code:

Authentication Code:

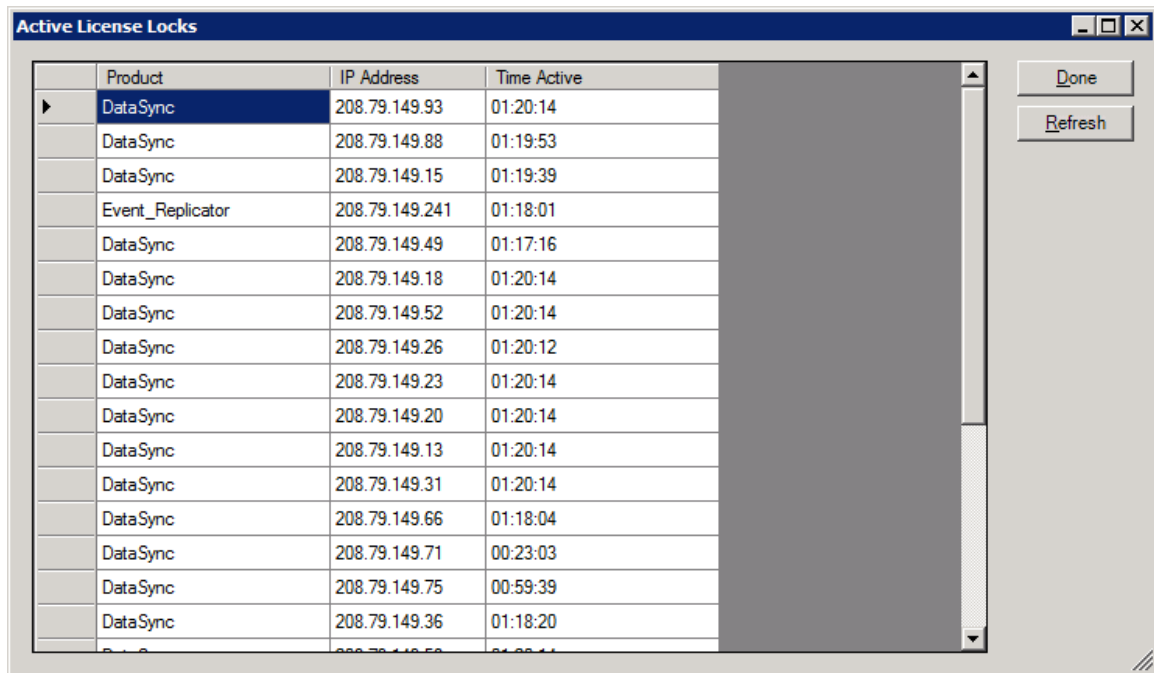
Activation Code:

888.88CONN | sales@connx.com | [Site Map](#) | Copyright 2013, CONNX Solutions, Inc. All rights reserved

7. Supply the Serial Number, Company Name, and License Code from the license certificate faxed or emailed as part of the CONNX package.
8. Supply the Authentication code and Hardware code you wrote down from the CONNX License Administrator Dialog from the computer that does not have internet access.
9. Press **Generate Activation**
10. Write down the **Activation Code** supplied.
11. Go back to the computer without internet access, that is running the License Administrator.
12. Fill in the activation code received from the web page (starts with **AV-**), and company name. Press Activate.
13. After several seconds, the **License Installed/Updated Successfully** window appears, confirming valid registration and activation.



14. To list all licenses currently in use, in the **CONNX License Administrator** window, click **View active license locks**.



The screenshot shows a window titled "Active License Locks" with a table of active licenses. The table has four columns: Product, IP Address, and Time Active. The first row is highlighted. To the right of the table are "Done" and "Refresh" buttons.

	Product	IP Address	Time Active
▶	DataSync	208.79.149.93	01:20:14
	DataSync	208.79.149.88	01:19:53
	DataSync	208.79.149.15	01:19:39
	Event_Replicator	208.79.149.241	01:18:01
	DataSync	208.79.149.49	01:17:16
	DataSync	208.79.149.18	01:20:14
	DataSync	208.79.149.52	01:20:14
	DataSync	208.79.149.26	01:20:12
	DataSync	208.79.149.23	01:20:14
	DataSync	208.79.149.20	01:20:14
	DataSync	208.79.149.13	01:20:14
	DataSync	208.79.149.31	01:20:14
	DataSync	208.79.149.66	01:18:04
	DataSync	208.79.149.71	00:23:03
	DataSync	208.79.149.75	00:59:39
	DataSync	208.79.149.36	01:18:20

15. Close the **CONNX License Administrator** window, or conduct any additional license upgrade procedures, as required.

Offline License Revoke

1. If the computer running the CONNX License Administrator does not have internet access, licenses must be revoked in "offline" mode, using the **Offline license revoke** button.
2. On the CONNX License Administrator, select the license to be revoked and press the Offline license revoke button.

The screenshot shows a dialog box titled "Offline License Revoke". It contains the following fields and buttons:

- Product:** Adabas
- Serial Number:** 223 [redacted] 091
- Hardware Code:** 879b3d83
- Authentication Code:** [empty field]
- Revoke Code:** [empty field]
- Buttons:** Revoke, Cancel
- Text:** The above code comes from <http://www.connx.com/revoke/Revoke.aspx>

3. Locate a computer that does have internet access, and go to <http://www.connx.com/revoke/revoke.aspx>.

The screenshot shows the "CONNX Offline License Revocation" web page in a Windows Internet Explorer browser. The page has a navigation bar with links: Login, Evaluate, Partners, About, Contact. Below the navigation bar is a search bar and a "Live Chat" button. The main content area is titled "CONNX Offline License Revocation" in large red text. Below the title are the following fields and buttons:

- Serial Number:** [empty field]
- Hardware Code:** [empty field]
- Company Name:** [empty field]
- Authentication Code:** [empty field] **Generate**
- Revoke Code:** [empty field] **Revoke**

At the bottom of the page, there is a footer with the text: 888.88CONN | sales@connx.com | [Site Map](#) | Copyright 2013, CONNX Solutions, Inc. All rights reserved.

The Serial Number and Hardware Code can be copied from the Offline License Revoke screen in step 2. Note: If you are unsure of the Company Name used for the license, it can be found on line 2 of the CONNX License file (CNXLIC.Lxx).

4. After filling in the Serial Number, Hardware Code and Company Name, press the Generate button to generate an Authentication code.

The screenshot shows a web browser window titled "CONNX Offline License Revocation - Windows Internet Explorer". The address bar shows the URL "http://www.connx.com/revoke/revoke.aspx". The page features the CONNX Solutions logo and navigation links: Login, Evaluate, Partners, About, Contact, Solutions, Products, Databases, Support, Learning Center, and Live Chat. The main heading is "CONNX Offline License Revocation". Below this, there are input fields for:

- Serial Number: 223 [redacted] 91
- Hardware Code: 879b3d83
- Company Name: CONNX Solutions, Inc.
- Authentication Code: 809 [redacted] 126, with a "Generate" button.
- Revoke Code: [empty], with a "Revoke" button.

 At the bottom, there is a footer with contact information: 888.88CONN | sales@connx.com | Site Map | Copyright 2013, CONNX Solutions, Inc. All rights reserved.

5. Copy the Authentication Code from the CONNX Offline License Revocation web page and paste it into the Authentication Code field on the License Administrator's Offline License Revoke screen.

The screenshot shows a dialog box titled "Offline License Revoke". It contains the following fields and buttons:

- Product: Adabas
- Serial Number: 223 [redacted] 91
- Hardware Code: 879b3d83
- Authentication Code: 809 [redacted] 126, with a message "Authentication code Validated." below it.
- Revoke Code: [empty]
- Buttons: Revoke, Cancel

6. After entering the Authentication Code, press the Revoke button to generate a Revoke Code.

The screenshot shows a dialog box titled "Offline License Revoke". It contains the following fields and buttons:

- Product:** Adabas
- Serial Number:** 223[redacted]91
- Hardware Code:** 879b3d83
- Authentication Code:** 8093[redacted]26
- Revoke Code:** RV-333[redacted]60
- Buttons:** Revoke, Close
- Message:** Authentication code Validated.
- Instructions:** Paste the above Revoke Code into: <http://www.connx.com/revoke/Revoke.aspx> to complete the revoke process.

7. Copy and paste the Revoke Code into the Revoke Code field on the CONNX Offline Revocation web page and press the Revoke button to complete the process.

The screenshot shows the "CONNX Offline License Revocation" web page in a Windows Internet Explorer browser window. The page displays the following information:

- Header:** CONNX Solutions logo, navigation links (Login, Evaluate, Partners, About, Contact), and a search bar.
- Menu:** Solutions, Products, Databases, Support, Learning Center, Live Chat.
- Section:** CONNX Offline License Revocation
- Status:** License has been successfully Revoked.
- Form Fields:**
 - Serial Number:** 223[redacted]91
 - Hardware Code:** 879b3d83
 - Company Name:** CONNX Solutions, Inc.
 - Authentication Code:** 8093[redacted]50126
 - Revoke Code:** RV33[redacted]9060
- Buttons:** Generate, Revoke
- Footer:** 888.88CONN | sales@connx.com | Site Map | Copyright 2013, CONNX Solutions, Inc. All rights reserved.

Note: All the steps outlined above must be completed. If the final revoke step on the web site is skipped, the license will be removed from the current license server but will not be able to be activated on a new license server.

Moving Licenses from one License Server to Another

If it becomes necessary to change the machine the License Server is running on, the registered licenses on the old machine will need to be revoked from that machine and then registered on the new machine.

To move the license files from one machine to another please do the following:

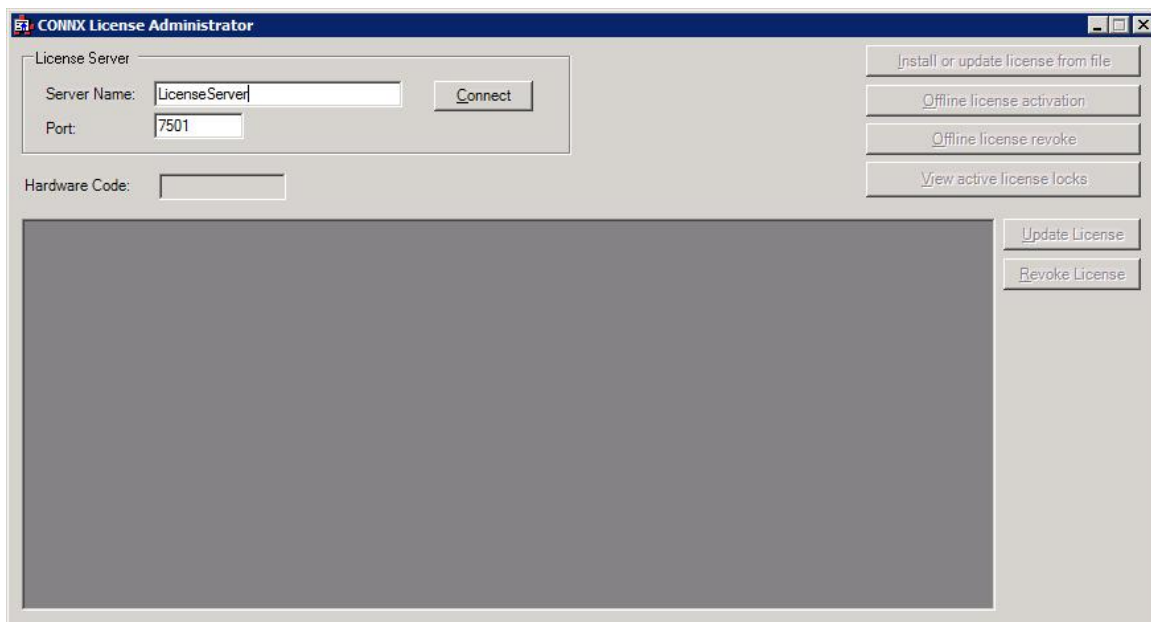
1. Install the license server on the new machine.
2. Revoke the license from the "old" system. This can be done 1 of 2 ways depending on how the license was originally registered:
 - a. If the license was registered with the [online registration process](#) (requires internet access)- then highlight the license file and select the "revoke license" button in the License Administrator.
 - b. If internet access is not available and the license was originally registered using the [offline registration method](#), the [offline license revoke method](#) will need to be used to revoke the license.
3. Register the license file(s) in the new license server:
 - a. If the new license server has internet access, use the [online registration process](#).
 - b. If the new license server does not have internet access, use the [offline registration process](#).
4. Alter all existing CONNX client installations to validate through the new license server.

Emergency License Activation

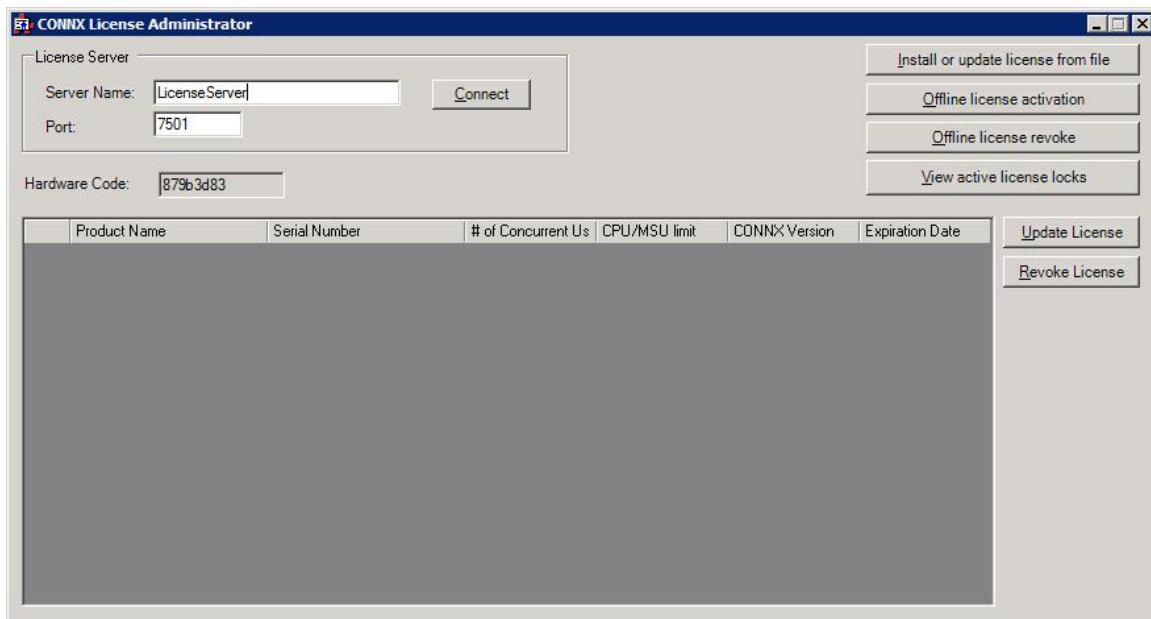
In the rare situation where licenses become invalid and not longer appear in the license administrator, and you are unable to contact technical support to resolve the issue (non-business hours, or weekend), a temporary 7 day license can be generated from <http://www.connx.com/Emergency/Emergency.aspx>

1. Locate a computer that does have internet access, and go to <http://www.connx.com/Emergency/Emergency.aspx>.

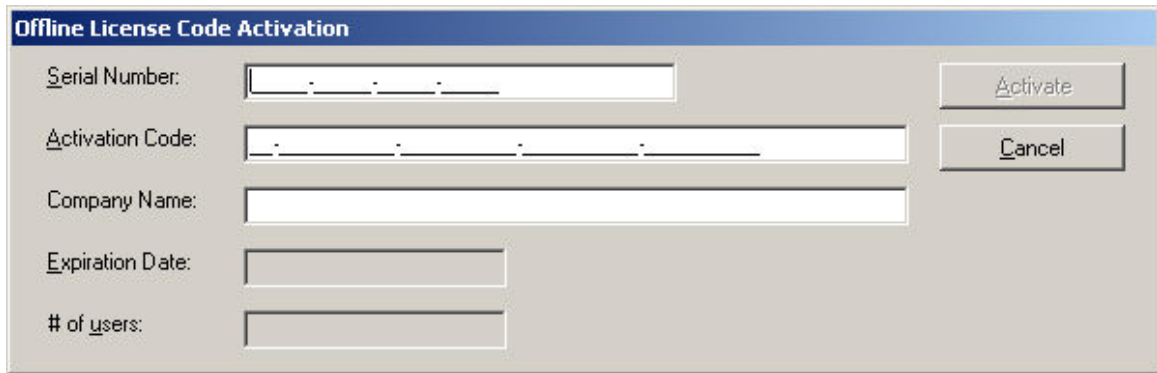
3. Supply the Serial Number, Company Name from the license certificate faxed or emailed as part of the CONNX package.
4. Supply the Hardware code that is reported in the CONNX License Administrator Dialog from the computer that does not have internet access.
5. Press **Generate Temporary 7 day License**.
6. Write down the **Activation Code** supplied.
7. On a CONNX client computer on which the CONNX Administrator component is installed, click **Start**, point to **All Programs**, point to **CONNX Driver** and then click **CONNX License Administrator**. The **CONNX License Administrator** window appears. The server name and port should be pre-populated with the correct CONNX license server, based on selections during installation. If the server name and port are not correct, enter the proper information. Press **Connect**.



8. Once connected to a license server, the **Install or Update license from file** button, **Offline license activation** button, **View active license locks**, and **Update license** button become enabled. Additionally, the **hardware code** of the license server (only required for off-line license activation) is displayed. If the computer running the license administrator has internet access, then proceed with online license activation by selecting either the **Install or Update license from file** button, or the **Update license** button.



9. Select the **Offline license activation** button.



Offline License Code Activation

Serial Number:

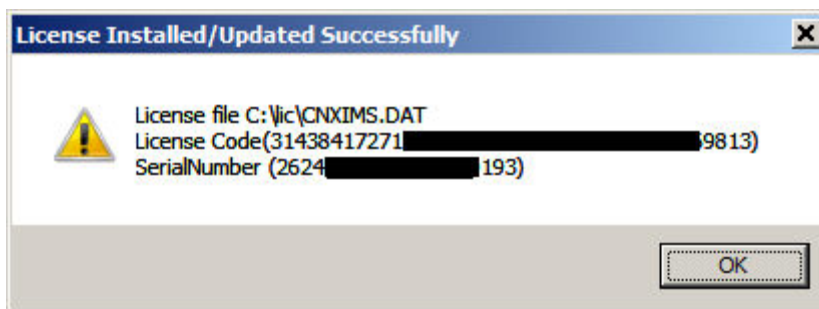
Activation Code:

Company Name:

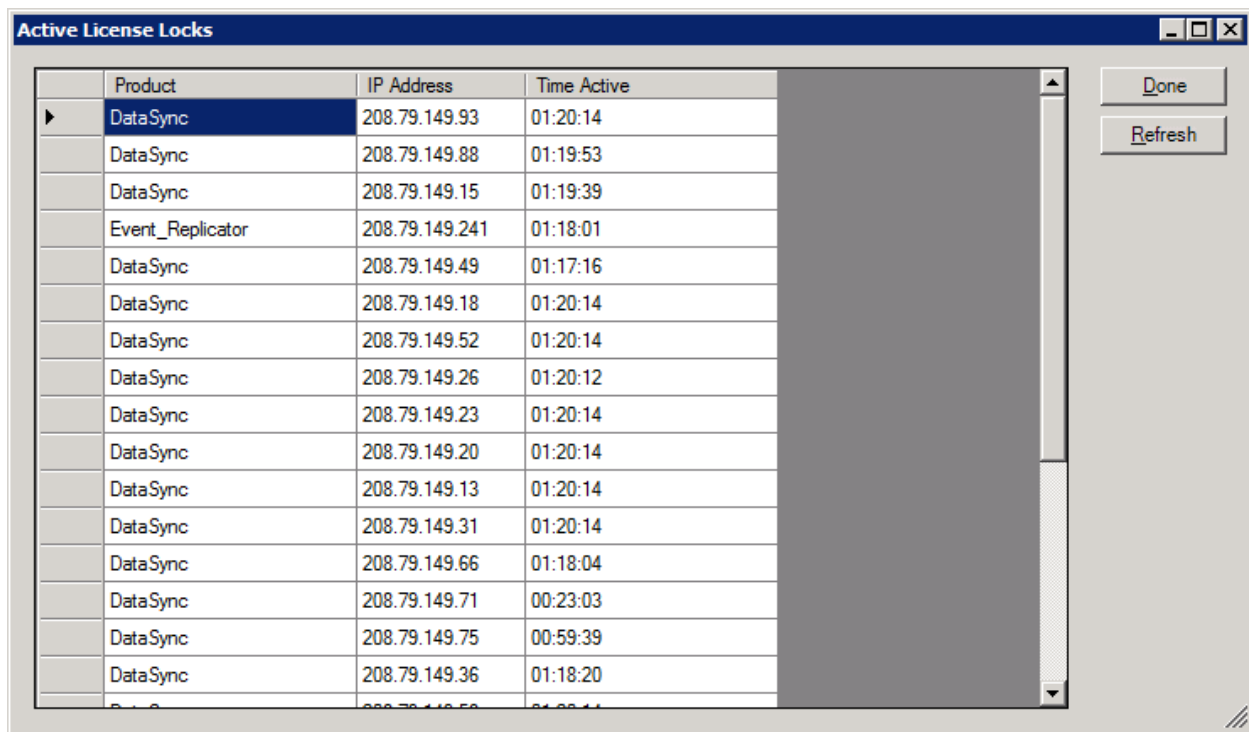
Expiration Date:

of users:

10. Supply the serial number, activation code (code received from web page .. starts with **AV-**), and company name. Press Activate.
11. After several seconds, the **License Installed/Updated Successfully** window appears, confirming valid registration and activation.



12. To list all licenses currently in use, in the **CONNx License Administrator** window, click **View active license locks**.



Active License Locks

	Product	IP Address	Time Active
▶	DataSync	208.79.149.93	01:20:14
	DataSync	208.79.149.88	01:19:53
	DataSync	208.79.149.15	01:19:39
	Event_Replicator	208.79.149.241	01:18:01
	DataSync	208.79.149.49	01:17:16
	DataSync	208.79.149.18	01:20:14
	DataSync	208.79.149.52	01:20:14
	DataSync	208.79.149.26	01:20:12
	DataSync	208.79.149.23	01:20:14
	DataSync	208.79.149.20	01:20:14
	DataSync	208.79.149.13	01:20:14
	DataSync	208.79.149.31	01:20:14
	DataSync	208.79.149.66	01:18:04
	DataSync	208.79.149.71	00:23:03
	DataSync	208.79.149.75	00:59:39
	DataSync	208.79.149.36	01:18:20

13. Close the **CONNX License Administrator** window, or conduct any additional license upgrade procedures, as required.

Frequently Asked Questions about Licenses

Can you have CONNX 11 and 12 clients connecting via the same license server?

Yes. A CONNX 11.x client can use a CONNX 12 license so as long as the license server has CONNX 12 licenses registered to it, both versions of the client will work. If the license server only contains CONNX 11 licenses, the CONNX 12 clients will not connect.

What happens if you try to install CONNX on a “server” operating system without an n-Tier license?

You will get an error message when you connect saying that an n-Tier license is required.

How can you install the license server on a “server” operating system without an n-Tier license when you have purchased a concurrent user license?

The license server does not require an n-tier license (or any other license).

Do you need to know the CPU count of the data server for an evaluation license?

No. CPU counts are not checked for evaluation licenses.

What happens if you install CONNX on a data server with a different CPU count?

Nothing happens at install time, it is only checked when you connect. When you connect however, if you are connecting to a server that has a higher CPU count than is specified in the license, an error will be returned.

How does CONNX check the CPU count in an OpenVMS cluster?

We use a VMS system call to check the CPU count.

How does CONNX check the MSU count on a Mainframe?

We use z/OS or z/VSE system calls to check the MSU count.

On a Mainframe, does CONNX use the MSU count for the entire system or just the LPAR where the CONNX Data Server is running?

We use the MSU count for the LPAR where the CONNX Data Server is running.

What happens if you try to activate a license more than once?

You will get a message indicating it is not permitted. If you wish to move the license to a different license server, you will first need to revoke it from the one it is currently activated on.

What happens if there is a problem with my license and I am unable to contact technical support to resolve the issue (non-business hours, or weekend)?

A temporary 7 day license can be generated. Please see the [Emergency License Activation](#) section for more information.

Upgrading and running multiple versions of CONNX

Backward and forward compatibility

In general, we always recommend that the client and server are at the exact same version number and build. However when upgrading CONNX in a large organization, or in situations where there are multiple environments, there may be a need to have multiple versions of CONNX active at the same time.

The CONNX Client

The CONNX client can only connect to a CONNX server that contains the server modules of the same major version number.

For example:

If the CONNX client is CONNX 14, and the CONNX server contains modules for CONNX 12 and CONNX 13, the connection will fail.

If the CONNX client is CONNX 14, and the CONNX server contains modules for CONNX 14 and CONNX 15, the connection will succeed.

The JDBC Jar File

The JDBC Jar file contains an embedded version file which contains the exact version and build number. The JDBC Jar file should only be used with JDBC servers with that exact build number or higher.

Using multiple version of CONNX simultaneously

In general, we always recommend that the client and server are at the exact same version number and build. However when upgrading CONNX in a large organization, or in situations where there are multiple environments, there may be a need to have multiple versions of CONNX active at the same time.

The CONNX Client

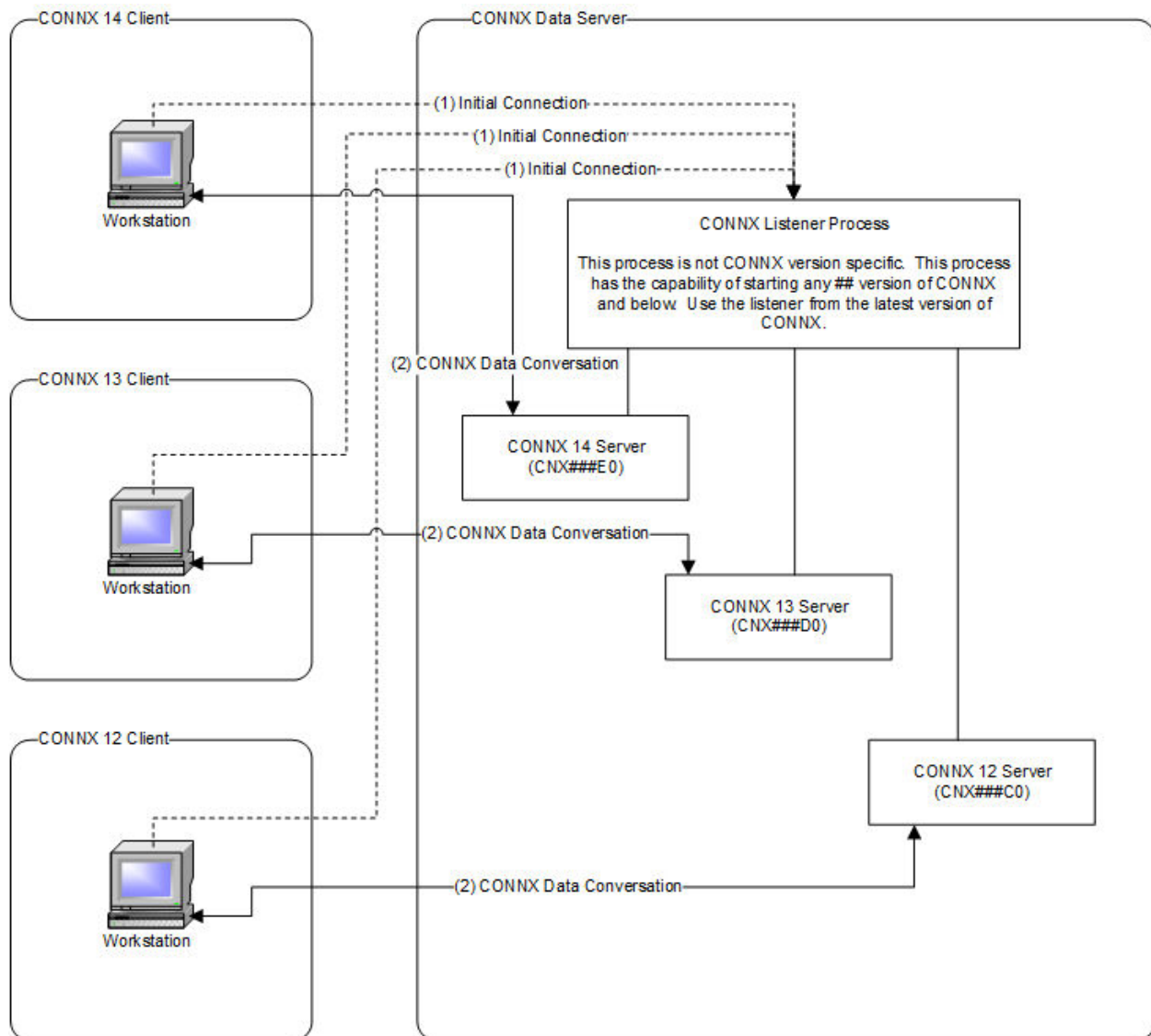
On windows there can only be one version of the CONNX client installed at any time. This is due do a legacy limitation of ODBC Drivers, where it was not easy to keep multiple versions of the same ODBC driver on the system without requiring application developers to change their code to point to a new driver name each time.

On Linux or unix, multiple versions of CONNX can coexist on the same system simply by installing into different directories. However the shared library path can only point to one version at any given time.

The CONNX Data Server & Listener

The CONNX Listener is designed to facilitate multiple major versions of CONNX simultaneously. The suffix of the server modules changes with each major version, allowing a customer to have a single mainframe, OpenVMS or unix TCP/IP port that can support connections from any client version.

To support a client, the matching server module must exist in the listener module area. On VMS, this means the server version executable must be installed to the same location. On Mainframe, the server modules for all of the required supported versions must be in the same data set. On Unix, the server executables need to be in the same directory as the listener cnxrun.



Silent CONNX Client Installation

Introduction to Silent Client Installation

Silent CONNX Client Installation allows for remote installation of the CONNX client. The key to a silent CONNX installation is a record file created during a record mode installation. The record file records the installer's responses and selections for the installation. Once the record file is created, it is used to repeat the installation procedure on multiple client computers.

This appendix describes the two procedures necessary to complete silent CONNX installation:

- [Creating the CONNX Client Installation Record File Procedure](#)
- [Running the Silent CONNX Client Installation Procedure](#)

Perform the first procedure to create the record file and then perform the silent CONNX client installation procedure to silently install the CONNX client.

Related Topics

 [CONNX Installation Procedure](#)

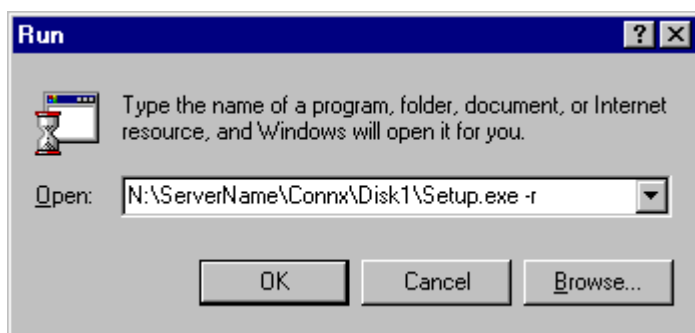
 [General Information for Installation](#)

Creating the CONNX Client Installation Record File Procedure

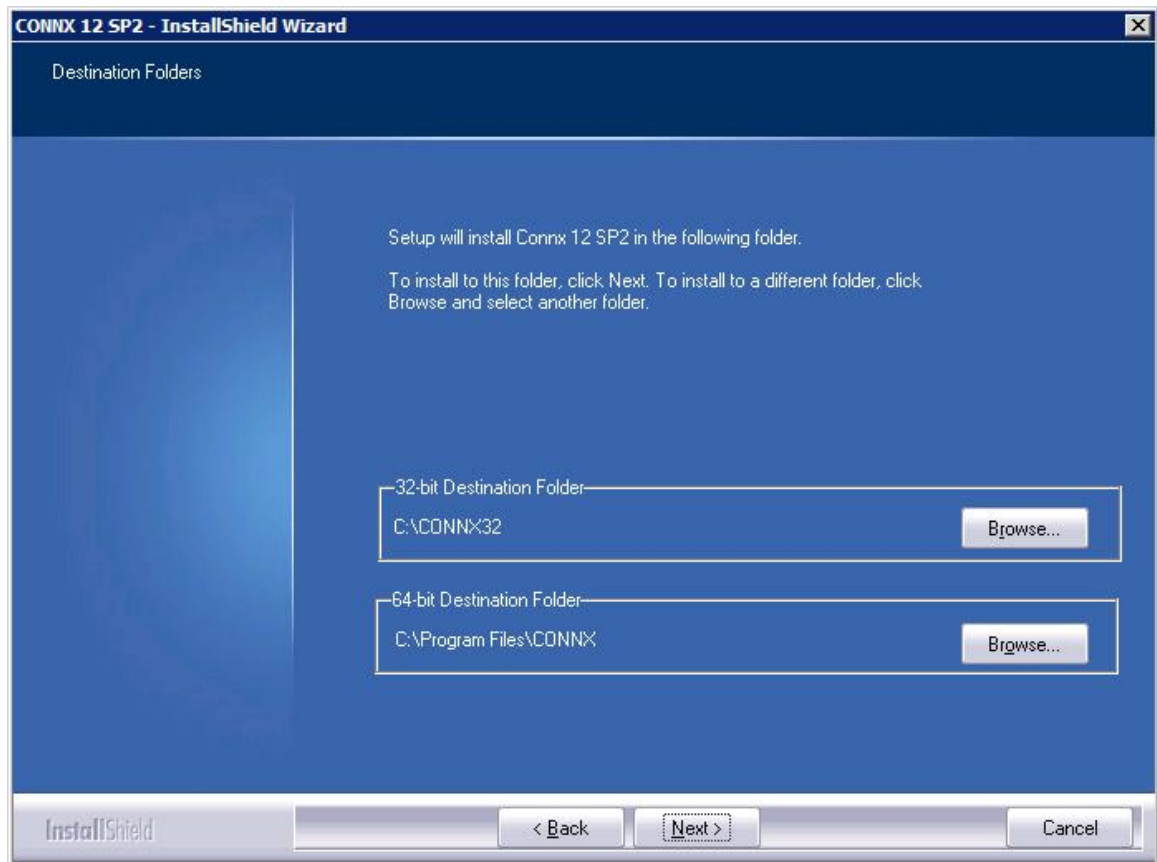
The following procedure describes how to create a CONNX client installation record file in preparation for silent CONNX client installation. This file serves as a template for future installations of CONNX.

To create the CONNX client installation record file

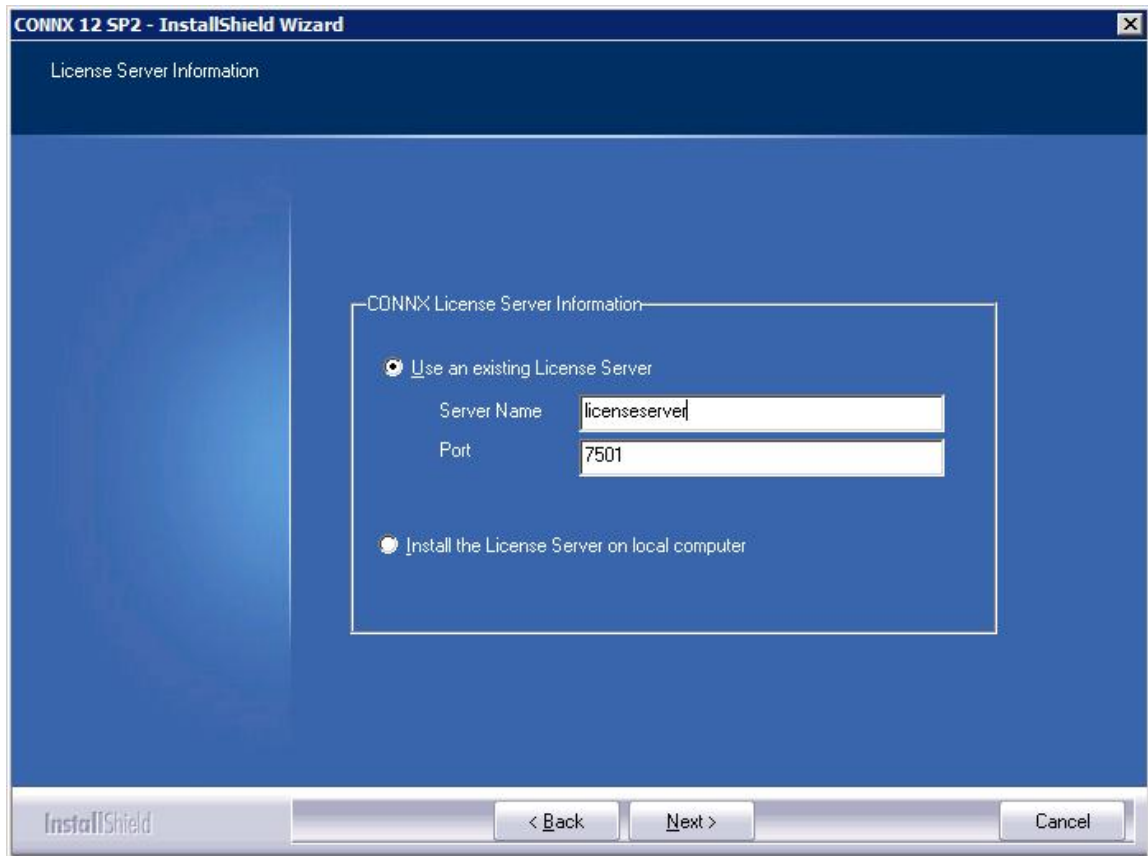
1. Copy the contents of the CONNX installation medium to a location on the network. Silent client setup is run from this location.
2. Select a client computer for the first silent installation.
3. Click **Start** on the taskbar, and then select **Run**. In the **Open** text box, type **N:\ServerName\Connx\Disk1\Setup.exe -r** where **N** represents a network drive. (A UNC path may be substituted for the network drive name; e.g., **\\UNCPath\CONN\Disk1\Setup.exe -r**).



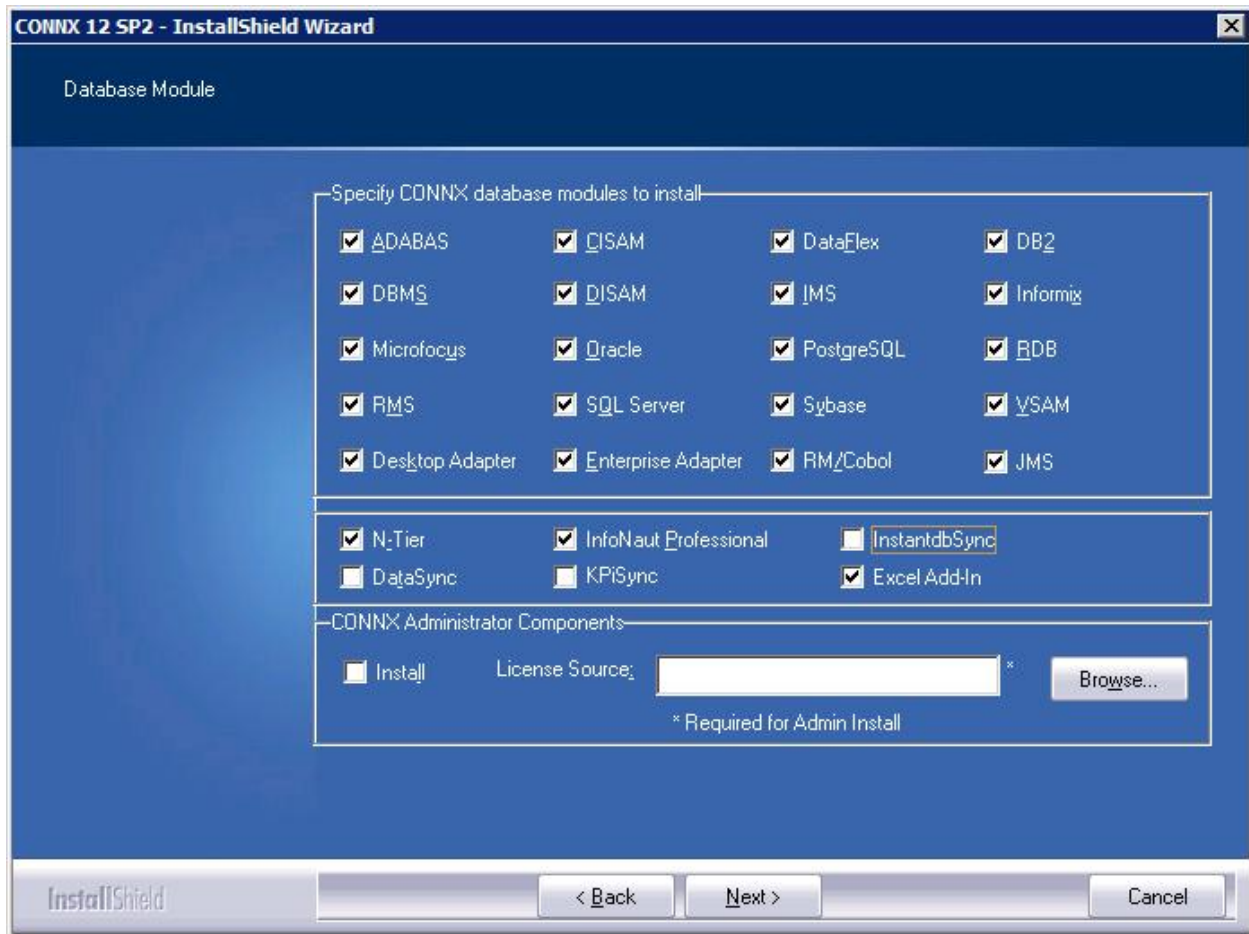
4. Click the **OK** button.
5. The Welcome to the InstallShield Wizard window appears. Click the **Next** button.
6. After you have read the License Agreement on the License Agreement screen, click the **I accept the terms of the license agreement** radio button, and then click the **Next** button.
7. The CONNX Location window appears. Click the **Browse** button to modify the destination folder for the program files. Click the **Next** button to continue.



8. The License Server Location screen appears. Click the **Use an existing License Server** radio button and enter the name of your CONNX license file license server. The default Port number is **7501**. Click the **Next** button to continue.

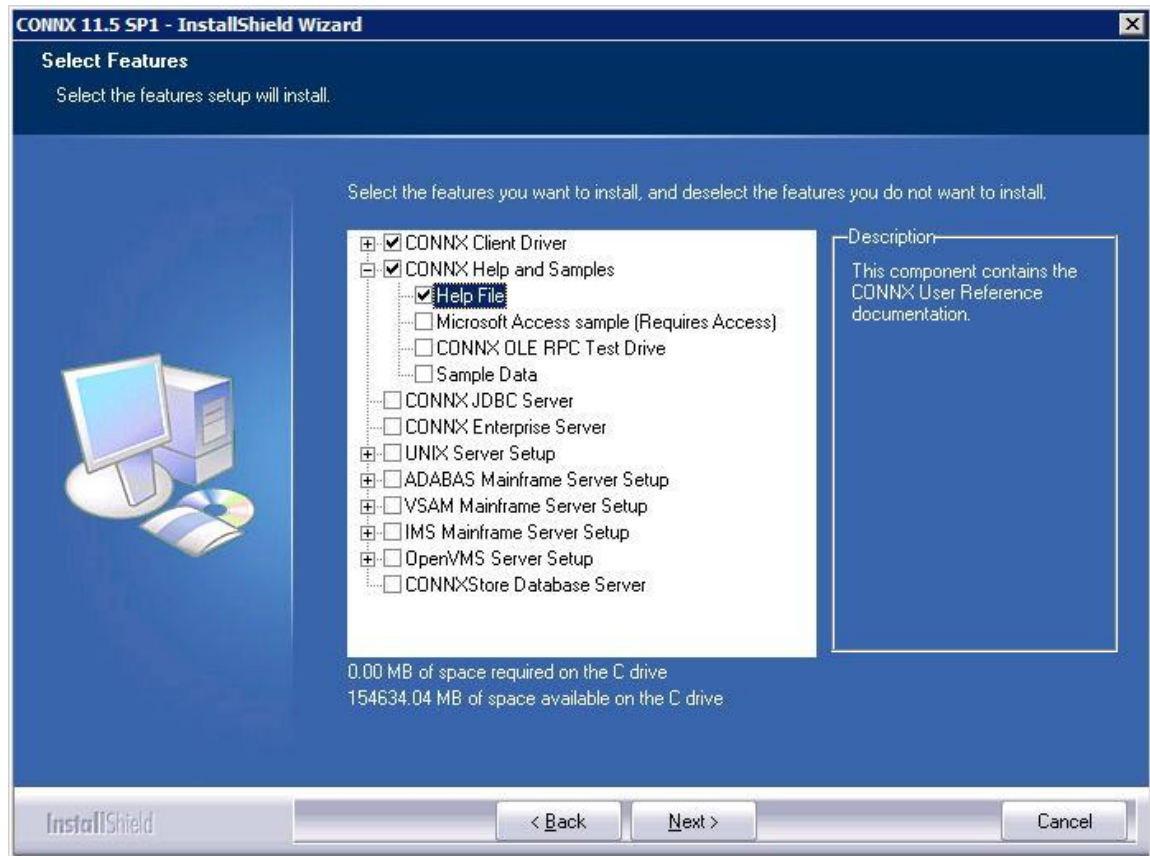


9. The Database Module screen appears. Select the databases for which you have a valid license code. If you are running databases other than VMS databases, specify the location of the license files. Do NOT check the **DataSync** check box. Uncheck the **Install** check box in the **CONNX Administrator Components** frame. Click the **Next** button to continue.

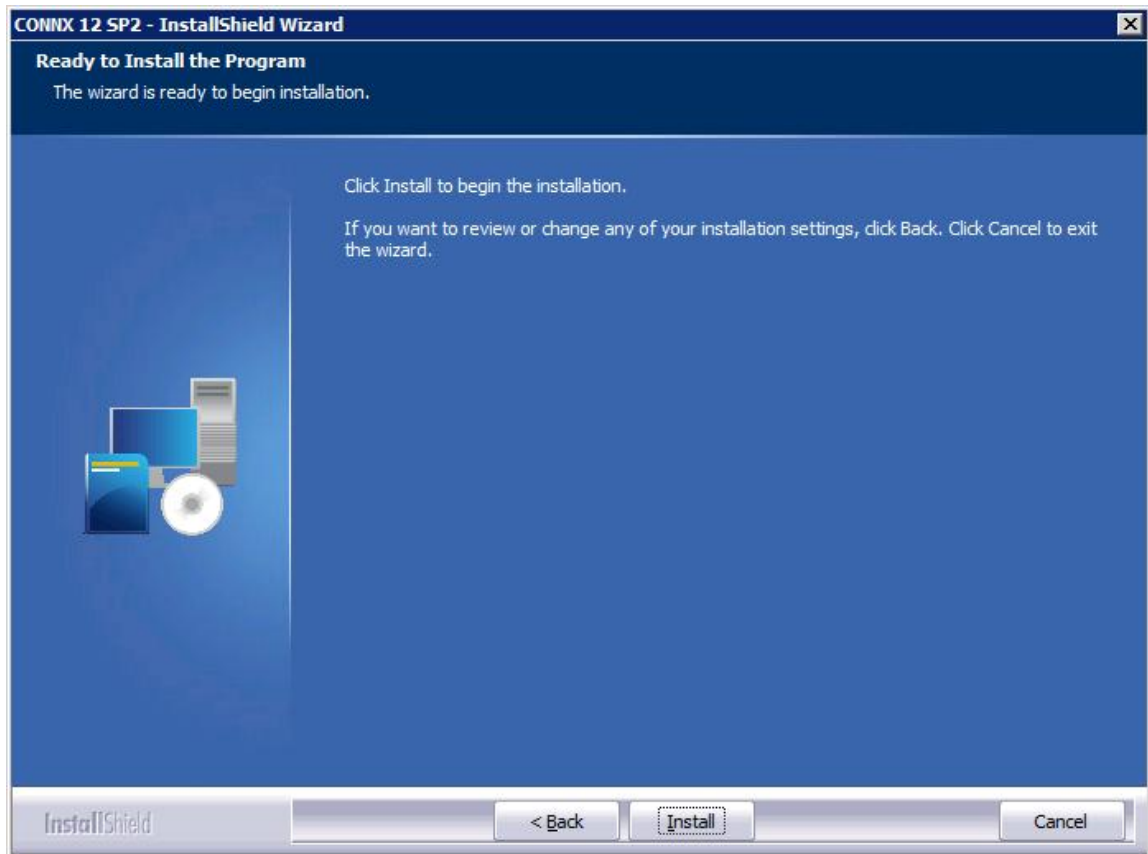


10. The Select Features window appears. The following should be checked:

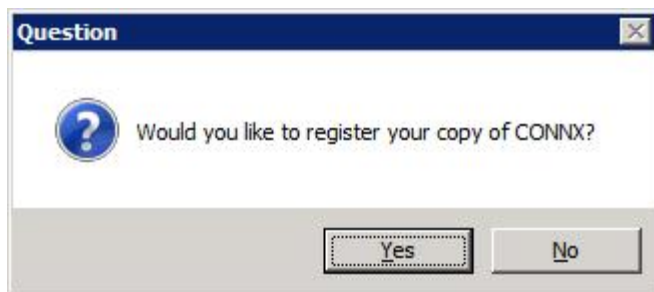
1. **CONNX Client Driver**
2. **CONNX Help File and Samples**
 - Click the plus sign to the left of the **CONNX Help File and Samples** to open the list of subcomponents available
 - Momentarily uncheck the **CONNX Help File and Samples** checkbox to uncheck all subcomponents (sample files are not available for client machines)
 - Check the **Help File** subcomponent. This will re-check the **CONNX Help and Samples** checkbox
 - Leave all other checkboxes under the **CONNX Help File and Samples** heading unchecked
3. Click the **Next** button to continue



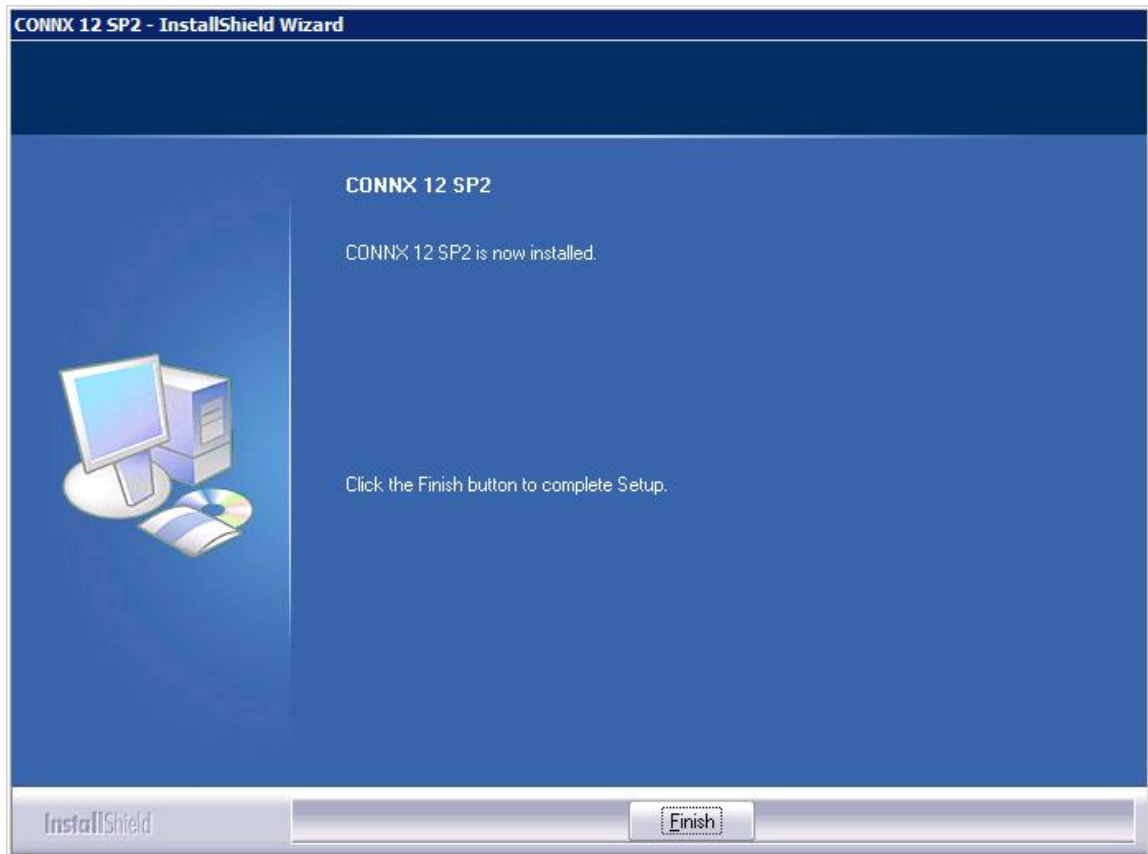
11. When the Ready to Install the Program screen appears, click the **Install** button to begin the installation.



12. Before setup completes, you are asked if you would like to register your copy of CONNX. Click **No**.



13. The Setup Complete window appears. Click the **Finish** button.



14. The record file you created is located at: **C:\WINDOWS\Setup.iss**. You **MUST** move the file to make it available for other silent installations. The most common location is a shared network location. Copy the record file C:\Windows\Setup.iss to a network server location; for example, N:\CNX\All\Setup.iss. You may also rename *.iss files to distinguish them from one another.

15. CONNX installation files. To facilitate client installation in instances where there are several or several hundred users, you may choose to copy all of the setup files from the CONNX installation medium into the same shared network folder as the *.iss files. A shortcut can then be sent to users via email. Each user could then execute the setup files for a silent client installation. See [To create a shortcut to CONNX setup](#) for instructions on how to create a shortcut.

16. Repeat this procedure for all configurations and types of CONNX installations.

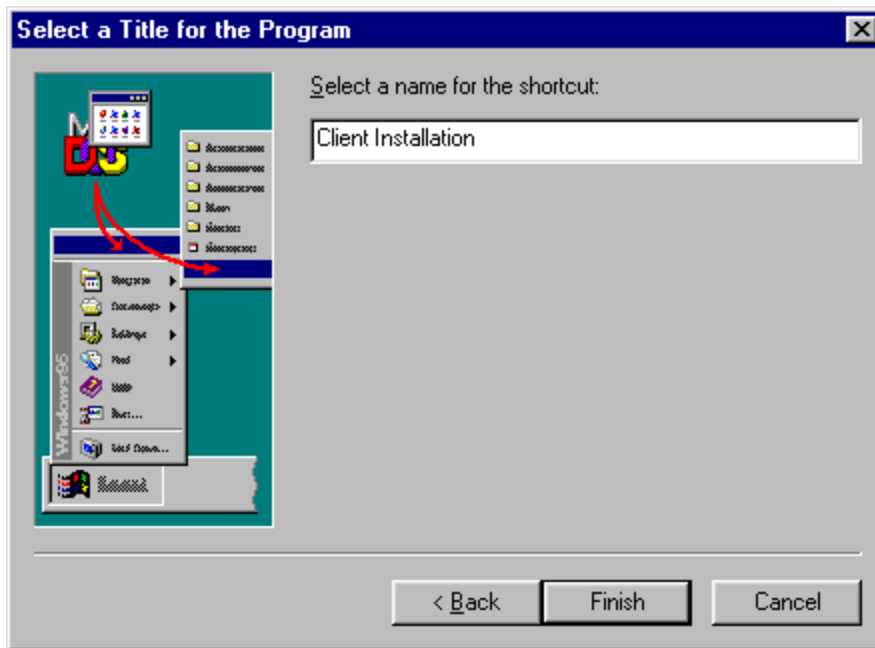
To Create a Shortcut to CONNX Setup

To expedite the silent client installation of CONNX, you can create a shortcut to the CONNX Setup.

1. The contents of the CONNX installation medium are stored in the same network share folder as the **Setup.iss** file, for example, N:\CNX\setup.iss.
2. On Windows Explorer, right-click on the **Setup.exe** file in the **N:\CNX** folder. Click **New**, select **Shortcut**, and then type the path to the setup files in the **Command Line** text box; for example, N:\CNX\setup.exe -s -f1N:\CNX\setup.iss

where N:\... represents the location in the network of the setup files.

3. Click the **Next** button and type a name for the shortcut; for example, **Client Installation**, and then click the **Finish** button.



4. The shortcut can then be sent via e-mail to all client machines for convenient installation of the silent client. In Windows Explorer, right-click on the shortcut, select **Send To**, select **Mail Recipient**, and enter the name of all recipients in the **To:** text box. Click **Send**.

Important: You must ensure that the network path you select is available to all client computers. If not, you must enter the full UNC path on each computer designated a CONNX client computer.

Note: If OLE DB or ODBC data sources are to be accessed by client machines, the driver or provider must be installed on ALL clients. If a System or User DSN is to be used, the SAME DSN must be set up on and made available to all clients.

Related Topics

- >> [Creating the CONNX Client Installation Record File Procedure](#)
- >> [Running the Silent CONNX Client Installation Procedure](#)

Running the Silent CONNX Client Installation Procedure

Conduct the following procedure to run a silent CONNX client installation on client computers. (See [Creating the CONNX Client Installation Record File.](#))

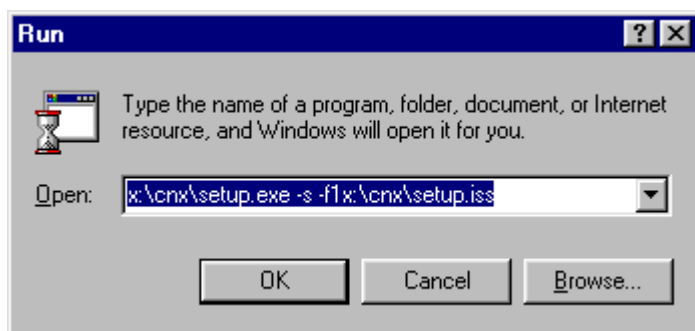
To run a silent CONNX client installation

1. Copy the shortcut you created from the network folder to a convenient location on the network.
2. Click the **Start** button, and then point to **Run**. In the **Open** text box, type the following:

N:\CNX\Setup.exe -s -f1N:\CNX\Setup.iss

where **N:\...** represents the location in the network of the setup files.

Note: Make sure you type a "1" (one) and not an "L" (el) after "-f".



3. Click the **OK** button
4. Allow several minutes for the installation to complete.
5. To verify installation of the silent client, click **Start**, point to **Settings**, click **Control Panel**, and then select **Administrative Tools**. Select **Data Sources (ODBC)**, and then select the **Drivers** tab. CONNX and its current version number should be visible.

About the SETUP.LOG file...

After a silent CONNX client installation is completed, a SETUP.LOG file is written to the same location of the Setup.iss Record File. Review this log and look for:

[ResponseResult]

ResultCode=0

where 0 indicates a successful installation. Other results include:

ResultCode=-1 General error

ResultCode=-2 Invalid mode

ResultCode=-3 Required data not found in the Setup.iss file

ResultCode=-4 Not enough memory available

ResultCode=-5 File does not exist

ResultCode=-6 Cannot write to the response file

ResultCode=-7 Unable to write to the log file

ResultCode=-8 Invalid path to the InstallShield Silent response file

ResultCode=-9 Not a valid list type (string or number)

ResultCode=-10 Data type is invalid

ResultCode=-11 Unknown error during setup

ResultCode=-12 Dialogs are out of order

ResultCode=-51 Cannot create the specified folder

ResultCode=-52 Cannot access the specified file or folder
ResultCode=-53 Invalid option selected

Related Topic

 [Creating the Silent Client Installation Record File Procedure](#)

Glossary

Terms and Abbreviations

ActiveX
Adabas
[ADO](#)
AIX
AnyNet
API
Applet
Application Requester (AR)
Application Server (AS)
APPC
Big-endian
Bind
CCSID
CICS
CMS
COM
Compiler
Conversation
Data Access Engine
Data set
Data set name
DB2
Database Management System (DBMS)
DBCS
DCT
DDM
DHTML
DOS/VSE
DRDA
DRDA connection
Dynamic SQL
ERP
ESDS
Extrapartition transient data queue
Heterogeneous data sources
HPUX
HTML
Java
JavaBeans

JavaScript
JCL
JCT
JDBC
JDK
JES
JNI
JVM
KSDS
LDS
Linux
Listener process
Little-endian
Localhost
Logical Unit (LU)
Logical Unit of Work (LUW)
Logical Unit Type 6.2 (LU 6.2)
MBCS
MDAC
MVS
MVS/ESA
ODBC
OLE DB
Package
PDS
PLT
Port
QSAM
RACF
RDB
Relational Database Name (Rdb Name)
RDO
RDS
recordset
RRDS
SBCS
SCO
SIT
SNA
Socket
Solaris
SQL

System/370

System/390

TCP/IP

Transaction program (TP)

trusted applet

TSO

UDA

UNC

Unit of work

UNIX

VM

VSAM

VSE

Well-known port

Windows DNA

z/OS

Index

3

390 system	165
390 target	252
390 z	245

A

Abbreviations	603
Activating	81
CONNX JDBC Router	81
Adabas . 106, 107, 339, 340, 342, 343, 344, 345, 347, 354, 363, 384, 520	
CONNX	106, 107, 384, 520
Adabas components	137
Adabas Host	141
CONNX	141
Adabas Installation Procedures	108
CONNX	108
Adabas on VSE	358
Adabas SAF Security	141
Adabas server startup procedure	382
edit	382
Adabas SQL Gateway.. 107, 326, 327, 335, 361, 380, 384, 520	
install	335
Prerequisites	326
Recommended Configurations	384
restart	380
Adabas SQL Gateway server component	368
install	368
Adabas Started Task TCP	127
Adabas TCP	337, 348, 349
Optional Entry Fields	337
Adabas Uninstallation	140
CONNX	140
Adabas user ID	121
Add SAS	291, 343
AIX Servers	419
verify server settings	419
APPC	464
Novell NetWare	464
APPN	464
Attachmate	473

tier SNA client configuration	473
-------------------------------------	-----

B

Barnard TCP	316, 356
enable	316, 356
Batch job	221
Batch Job SCT metadata import functionality	202
Batch Job TCP	199, 209
Batch Job User ID	206
Build C	76
Build CONNX DB2 sample tables	53
target host via TCP	53
Build Desktop Informix sample database tables	72
Build Desktop OLE DB sample database tables	62
Build Desktop Sybase sample database tables	74
Build Enterprise ODBC sample database tables	60
Build Enterprise OLE DB sample database tables	57
Build SQL Server ODBC sample database tables	70
Build SQL Server OLE DB sample database tables	67

C

C 403	
restart	403
C transient data queue definitions	291, 343
CICS Startup JCL	291, 343
C®	390, 401, 513
edit	401
General Information	390
Change ..214, 216, 255, 260, 307, 308, 353, 354	
CICS startup JCL RPL list	255
CONNX environment variable	260, 308, 354
CONNX persistent environment variable ...	214
number	216, 307, 353
Choose	553, 554
Output Type	553
Query Type	554
CICS	242, 263, 294, 345

CONNX.....	242	Installing	156
Installed Components.....	294, 345	CONNX DB2 Dynamic SQL Packages.....	404
CICS Customization.....	254, 287, 342	CONNX DSN Prefix Data Sets	249
Completing	254, 287, 342	CONNX environment variable	260, 352
CICS DCT	288	change.....	260
CICS extrapartition data queues.....	288	set.....	352
CICS Startup JCL.....	291, 343	CONNX Environment Variables....	220, 259, 306, 308, 310, 354
C transient data queue definitions.....	291, 343	change.....	308, 354
CICS startup JCL RPL list.....	255	set.....	259, 306
change	255	CONNX IMS components.....	162
CICS/VSE 2.3 External Security Interface	285	Uninstalling.....	162
Client	21	CONNX Installation.....	48, 50
Client Security Overview	141	Completing	48, 50
COBOL copybooks	181	CONNX Installation Checklist	4
Compatible Server	519	CONNX Installation Procedure	30, 40
Completes	48, 50, 254, 287, 342	CONNX JDBC Applet	559
CICS Customization	254, 287, 342	run	559
CONNX Installation	48, 50	CONNX JDBC Client Requirements.....	26
Configuration	418	CONNX JDBC General Information.....	78
support.....	418	CONNX JDBC Router.....	81
Configuration Utility	296, 347	Activating.....	81
Configure.....	464	CONNX JDBC Sample Applet.....	558
node.....	464	Running	558
Configure Data Link Adapters.....	466	CONNX JDBC Sample Application.....	546, 548, 549, 551
Configure DDF	497	Running	551
TCP.....	497	Using	548, 549
Connect.....	504, 510	CONNX JDBC Server.....	79, 80, 82, 84, 514
DB2.....	504	Connecting	82
target DB2 region via CONNX.....	510	Stopping	80, 84
Connected!.....	82	CONNX JDBC Server Requirements	27
Connection Problems.....	540	CONNX JDBC Works	8
Troubleshooting.....	540	CONNX Library	267, 327
CONNX CICS.....	297, 304, 348, 349	Creating.....	267, 327
start.....	297, 348	CONNX Listener process	115
stop	304, 349	start	115
CONNX CICS Components	256, 292, 344	CONNX OpenVMS Installation Procedure	94
Installation Verification.....	256, 292, 344	CONNX persistent environment variable.....	214
CONNX Client Installation Record File		change.....	214
Procedure	593	CONNX port number	261, 309, 355
Creating	593	convert.....	261, 309, 355
CONNX components.....	156, 165, 245		
install.....	165, 245		

CONNX Registry File Settings	517	display	262
Modifying	517	Current environment variable	317, 357
CONNX sample customer partitioned data set	193	display	317, 357
import metadata.....	193	D	
CONNX Sample Database Tables.....	51	Data Dictionaries.....	538
Creating	51	Converting	538
CONNX sample QSAM customer physical sequential file.....	187	Database Connection	531
import metadata.....	187	Testing.....	531
CONNX Server.....	334	Database Terminology.....	2
CONNX setup	600	DataFlex.....	516
CONNX Started Task.....	199, 223	DataFlex Year Digits.....	517
stop	199	Dataset Security.....	236
CONNX Started Task components	176, 230	DB2	504, 518
verify installation	176	connect.....	504
CONNX Started Tasks	209	OS	504
stop	209	DB2 Admins	410
CONNX System Requirements.....	15	Host Configuration	410
CONNX TCP	216, 258, 307, 353	DB2 CDD configuration options.....	408
start.....	258	DB2 DDF.....	494
CONNX TSO Configuration Utility	208, 211	DB2 Transaction Program Name.....	437
Introduction.....	208	verify.....	437
CONNX Upgrades.....	515	DBMS Databases	537
CONNX VSAM Server	243, 274	Default.....	218, 261, 309, 355
CONNX Works	7	Default Port.....	28
Connxserver shell script.....	381	Define unique CONNX environment variables per Started Task.....	221
edit.....	381	Defines	436, 453, 467, 470, 471, 494
Control VSAM file Enables.....	310	mode	471
Convert.....	218, 261, 309, 355, 538	mode name	436
CONNX port number	261, 309, 355	peer link.....	467
Data Dictionaries	538	RDBName	494
persistent CONNX port number	218	server node	453
Create.....	51, 267, 327, 501, 593, 600	Transaction Program Name	470
CONNX Client Installation Record File Procedure	593	Determine	136
CONNX Library.....	267, 327	TCP	136
CONNX Sample Database Tables	51	DISAM.....	390, 391, 401, 403
shortcut.....	600	Display	6, 262, 317, 357
SNATCP macro.....	501	current CONNX environment variables.....	262
Create Oracle sample database tables.....	52	current environment variable.....	317, 357
Current CONNX environment variables.....	262	Your CONNX Version	6
		DLC.....	488
		DLC configuration	430

DLC scenario	442, 463	Import functionality against SCT VSAM master files	200
DLI	154	Import metadata	181, 187, 193
Docker	44, 374	CONNX sample customer partitioned data set	193
DRDA connections	436	CONNX sample QSAM customer physical sequential file	187
DSN Prefix Data Sets	169	IMS	147, 154, 249, 252, 254, 261, 262
E		Prerequisites	147, 154
Ease	222	IMS DSN Prefix Data Sets	160
Use	222	IMS Installation JCL	161
Edit	87, 381, 382, 401, 528	IMS Server Setup	158
Adabas server startup procedure	382	Optional Entry Fields	158
C 401		Informix data sources	521
connxserver shell script	381	Install	156, 165, 245, 252, 275, 281, 335, 339, 368, 391
JDBC server startup procedure	87	Adabas SQL Gateway	335
OpenVMS Site	528	Adabas SQL Gateway server component ..	368
Emergency License Activation	584	CONNX	275
Enable	316, 356	CONNX C	391
Barnard TCP	316, 356	CONNX components	156, 165, 245
Enable RACF security	120	selected components	252, 281, 339
Environment Variable Security vs	222	Install Desktop ODBC sample database tables	65
Establish CONNX	408	Installation	10, 29, 39
Establish RDBName configuration	421	General Information	29, 39
Establish TCP	411, 423	Preparing	10
G		Installation Checklist	512
General Information ... 11, 29, 39, 102, 243, 274, 334, 367, 390		Installation Guide	1
C 390		Installation Overview	3
Installation	29, 39	Installation Procedures	103
General Information Regarding	92, 512	Outline	103
OpenVMS Server	92	Installation Verification	175, 256, 292, 344
Post	512	CONNX CICS Components	256, 292, 344
General Information Regarding License Registration	562	Introduction	175
H		Installed Components	294, 345
Host Configuration	410	CICS	294, 345
DB2 Admins	410	Introduction	112, 172, 175, 208, 230, 592
Host name	502	CONNX TSO Configuration Utility	208
I		installation verification	175
IBM CICS TCP	257	Silent Client Installation	592
start	257	Started Task installation procedures	172
IBM Com Server	432	uninstallation	230
IBM Mainframe	519		
IBMRDB	436		

- IP53, 101, 497, 542, 543, 544
- IP configuration423
- IP configuration on DB2 UDB411
- IP Listener 127, 199, 209, 258, 297, 304, 348, 349
- IP port.....307, 353
- IP port persistent environment variable216
- IP Server Setup 167, 247, 277, 337
- IP settings.....414
 - Linux414
- IP settings via DB2 CLP.....413
- IP sockets interface.....257
- IP stack.....136, 316, 356
- ISAM..... 390, 391, 401, 403
- ISAM sample database tables76
- ISAM, DISAM513
- J**
- Jar File.....89
- JDBC server startup procedure87
 - edit.....87
- L**
- License
 - Activation563, 571
 - FAQ588
 - Offline571, 577
 - Revoke.....563, 577
 - Upgrade Registration.....563
 - View License Locks563
- Linux.....414
 - IP settings.....414
- LOG Using FTP544
- M**
- Mainframe387
- Mainframe Installation366
- Manual Steps 140, 235, 321, 361
- Micro Focus.....390, 513
- Micro Focus component.....391
 - Unix machine391
- Micro Focus server.....403
- Micro Focus server startup procedure401
- Microsoft Windows Environment.....548
- Minimum OpenVMS Quotas Procedure.....535
- Reviewing.....535
- Minimum Quota Values537
- RMS537
- Mode471
 - define.....471
- Mode name436
 - define.....436
- Modify517
 - CONNX Registry File Settings517
- MPTN473
- MPTN configuration425
- MPTN gateway432
- Multiple instances223
 - Starting223
- Multiple Versions.....97
 - Running97
- MVS504
- N**
- Navigator Bar557
- Node464
 - configure.....464
- Non549
- Novell NetWare.....464
 - APPC.....464
- Number216, 307, 353
 - change.....216, 307, 353
- O**
- ODBA.....147
- Open310
- OpenVMS542
- OpenVMS Server.....92, 97, 526, 543
 - CONNX97
 - General Information Regarding.....92
 - Pinging543
- OpenVMS Server Installation99
 - Testing.....99
- OpenVMS Site528
 - Editing528
- OpenVMS User Account.....93
- Optional Entry Field 158, 167, 247, 277, 337
 - Adabas TCP337
 - IMS Server Setup158

VSAM TCP	167, 247, 277
Oracle	522
OS	165, 242, 252, 263, 504
DB2	504
OS system	245
OS Uninstallation Without Shutting Down CICS	263
Outline	103
Installation Procedures	103
Output Type	553
choose	553
P	
Parameter query	555, 556
Password	93, 206, 244
Password verification	121
PDS	164
Peer link	467
define	467
Persistent CONNX environment variable	212
set	212
Persistent CONNX port number	218
convert	218
Persistent environment variables	211
Pinging	543
OpenVMS Server	543
Port	28
Post	512
General Information Regarding	512
Post-Installation Checklist	521
Post-installation checklists ...	513, 514, 515, 516, 518, 519, 520, 522, 524, 525, 526, 530
Post-Installation Procedure	83
POWERflex	524
Precedence	220
Pre-Installation Checklist Procedure	12
Preparing	10
Installation	10
Prerequisites	107, 147, 154, 164, 242, 266, 326, 387, 388, 389
Adabas SQL Gateway	326
IMS	147, 154

Q	
QSAM	164
Query Type	554
choose	554
R	
Rdb	537
RDBName	494
define	494
Recommended Configurations	384
Adabas SQL Gateway	384
References	511
Required	158, 167, 247, 277, 337
Restart	380, 403
Adabas SQL Gateway	380
C 403	
Reviewing	535
Minimum OpenVMS Quotas Procedure	535
RMS	537, 541
Minimum Quota Values	537
RUMBA MPTN	483
tier SNA client configuration	483
Run	97, 263, 323, 363, 551, 558, 559, 601
CONNX	263, 323, 363
CONNX JDBC Applet	559
CONNX JDBC Sample Applet	558
CONNX JDBC Sample Application	551
Multiple Versions	97
Silent CONNX Client Installation Procedure	601
S	
SAMPLES	170, 250, 279
SAS/C Network Administration	135
Security	106, 141
Adabas SAF Security	141
Selected components	252, 281, 339
install	252, 281, 339
Server node	453
define	453
Set	212, 259, 306, 352, 425, 432, 473, 483
CONNX environment variable ...	259, 306, 352
persistent CONNX environment variable ...	212
Settings via Command Line Processor	417

Settings via Control Center	415	determine	136
Setup	285, 503	Terms	603
Shortcut	600	Test	99, 100, 121, 206, 531
create	600	CONNX	121, 206
Silent client installation	592	Database Connection	531
Introduction	592	OpenVMS Server Installation	99
Silent CONNX Client Installation Procedure ..	601	User Account Privileges	100
Running	601	Test CONNX	200, 202
SNATCP macro	501	VSAM query	200
create	501	VSAM Started Task	202
Sources	511	Tier SNA client configuration	473, 483
Specific Startup Command	528	Attachmate	473
Start	115, 223, 257, 258, 297, 348	RUMBA MPTN	483
CONNX CICS	297, 348	Transaction Program Name	470
CONNX Listener process	115	define	470
CONNX TCP	258	Troubleshooting	540
IBM CICS TCP	257	Connection Problems	540
multiple instances	223	U	
Start_JDBC_Server_UNIX	84	Uninstall	231
Started Task installation procedures	172	CONNX	231
Introduction	172	Uninstall Adabas SQL Gateway	359
Stop ... 80, 84, 127, 199, 209, 304, 349, 380, 403		Uninstall CONNX	319
CONNX	127	VSAM	319
CONNX CICS	304, 349	Uninstallation	230
CONNX JDBC Server	80, 84	Introduction	230
CONNX Started Task	199, 209	Uninstallation Procedures	318, 358
Summary	282, 340	Uninstalling	137, 162
CONNX	282, 340	CONNX	137
Supports	418	CONNX IMS components	162
configuration	418	Unix	40, 84, 87, 382, 389
Sybase data sources	525, 530	Unix Client System Requirements	22
System Requirements Checklist	13	Unix Environment Variables	379
T		UNIX JDBC Servers	50
Target CICS	245	Unix machine	368, 391
Target DB2 region via CONNX	510	Micro Focus component	391
connect	510	UNIX ODBC Drivers	48
Target host via TCP	53	Unix System	39
build CONNX DB2 sample tables	53	Upgrade Installation Checklist	5
Target IMS system	156	Use	222
Target MVS	165	Ease	222
TCP	101, 136, 497, 542, 543, 544	User Account Privileges	100
configure DDF	497	Testing	100

User Application Requirements	24	VSAM Started Task components.....	231
Using	548, 549	VSAM Started Task Uninstallation.....	235
CONNX JDBC Sample Application	548, 549	CONNX	235
Using CONNX with unixODBC	83	VSAM TCP.....	167, 247, 277, 297, 304
Using Your Telnet Utility.....	101	Optional Entry Fields.....	167, 247, 277
V		VSAM User Account	244
Verify	437	VSE	266, 319, 326, 327, 359
DB2 Transaction Program Name	437	VSE components	275, 335
Verify correct IP address.....	502	VSE Installation Procedure.....	282, 340
Verify IBMRDB mode description	431	VSE Partition.....	294, 345
Verify installation	115, 176	VSE target.....	281, 339
CONNX Started Task components	176	VSE Uninstallation	321, 361
Verify MPTN APPC Controller configuration..	427	CONNX	321
Verify server settings.....	419	VSE Uninstallation Without Shutting Down CICS 323, 363
AIX Servers	419	W	
Verify server TCP	413, 414	Web Server Requirements	21
Verify System Requirements Procedure.....	14	Windows	388
VSAM	164, 242, 263, 266, 275, 282, 319	Windows Environment	549
CONNX.....	164, 266	Windows Systems.....	80
uninstall CONNX	319	Y	
VSAM OS	245	Your CONNX Version	6
VSAM query	200	Displaying.....	6
test CONNX.....	200	Z	
VSAM Started Task.....	202, 206	Z 242, 263	
test CONNX.....	202		
