IBM Visual Warehouse for Windows NT

Using Classic Connect with Visual Warehouse

Version 5 Release 2

SC26-9857-00

IBM Visual Warehouse for Windows NT

Using Classic Connect with Visual Warehouse

Version 5 Release 2

SC26-9857-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 41.

First Edition (July 1999)

This edition applies to Version 5 Release 2 of Visual Warehouse (5639–VW5) and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

Contents

About this book

This book explains how to use Classic Connect with Visual Warehouse™.

Use this book with *DataJoiner® Classic Connect: Installation, Configuration, and Reference Guide* and *Managing Visual Warehouse*.

Who should use this book

You should read this book if you are responsible for setting up access to Classic Connect data servers and enterprise servers from Visual Warehouse. You should be familiar with installing and configuring TCP/IP and SNA communications software.

How to send your comments

Your feedback is important in helping to provide accurate and high-quality information. If you have any comments about this book or any other Visual Warehouse documentation, visit the following Web site:

`http://www.software.ibm.com/data/vw`

There you will find a feedback page where you can enter and submit your comments.

Chapter 1. What is Classic Connect?

Classic Connect provides read access to nonrelational data stored in Information Management Systems (IMS™) databases and Virtual Storage Access Method (VSAM) data sets on MVS™. It provides communication, data access, and data mapping functions so you can read nonrelational data using relational queries.

This chapter contains the following sections:

- "What does it do?"
- "Which data sources does it access?"
- "How is it used?" on page 2
- "What are its components?" on page 2

What does it do?

Classic Connect allows you to access nonrelational data by issuing a standard SQL query from a Visual Warehouse business view. You access the data just as if the data was in a DB2® database.

Which data sources does it access?

Classic Connect provides read-only relational access to IMS databases and VSAM data sets. It creates a logical, relational database, complete with logical tables that are mapped to actual data in IMS or VSAM databases. Specifically:

- For VSAM, each logical table corresponds to a VSAM data set. Each field of the data set maps to a column of the logical table; each record of the data set maps to a row. Classic Connect can read data from VSAM KSDS, RRDS, and ESDS data sets.
- For IMS, each logical table corresponds to one or more segments of a path in an IMS full-function database. The fields of multiple segments in a path correspond to the columns of a logical table. Each unique set of segment instances for the given path corresponds to a row in a logical table.

Using this relational structure, Classic Connect interprets relational queries that are submitted by users against IMS databases and VSAM data sets.

You can define multiple logical databases for a single data source (such as a set of VSAM data sets or a single IMS database). Multiple logical tables can be defined in a logical database.

You can define multiple logical tables for a single data entity (such as a VSAM data set or an IMS segment). For example, a single VSAM data set can have multiple logical tables defined for it, each one mapping data in a new way.

How is it used?

Use Classic Connect with Visual Warehouse if your data warehouse uses operational data in an IMS or VSAM database. Use Classic Connect to map the nonrelational data into a pseudo-relational format. Then use the CROSS ACCESS ODBC driver to access the pseudo-relational data. You can then define an IMS or VSAM information resource in Visual Warehouse that corresponds to the pseudo-relational data.

What are its components?

Using Classic Connect with Visual Warehouse consists of the following major components:

- "Visual Warehouse agents" on page 3
- "CROSS ACCESS ODBC driver" on page 4
- "Classic Connect data server" on page 4
- "Enterprise server" on page 7
- "Data mapper" on page 9

Figure 1 on page 3 shows how Classic Connect and its components fit into the overall Visual Warehouse architecture.

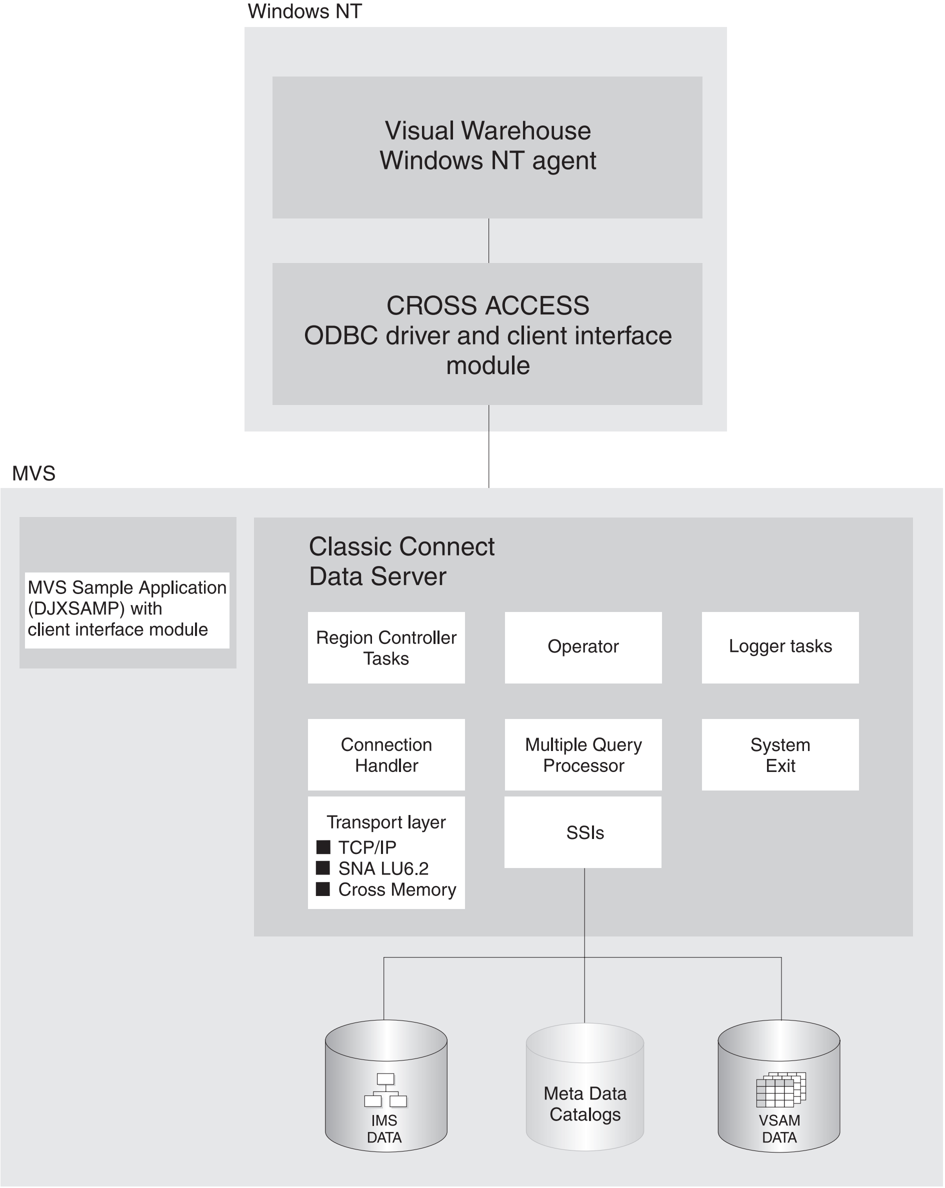


Figure 1. Classic Connect architecture

Visual Warehouse agents

Visual Warehouse agents manage the flow of data between the data sources and the target warehouses. The agents use the CROSS ACCESS ODBC Driver to communicate with Classic Connect.

CROSS ACCESS ODBC driver

The Open Database Connectivity (ODBC) interface enables applications to use Structured Query Language (SQL) statements to access data in relational and nonrelational database management systems.

The ODBC architecture consists of four components:

- The ODBC-compliant application performs processing and calls the ODBC functions to submit SQL statements and retrieve results.
- The Driver Manager loads drivers on behalf of an application.
- The driver processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application.
- The data source consists of the data the user wants to access. The data source name is equivalent to the data source name in the CROSS ACCESS data server configuration file.

The Driver Manager and the driver act as one unit that processes ODBC function calls.

Classic Connect data server

All data access is performed by Classic Connect data servers. A data server is responsible for the following functions:

- Accepting SQL queries from Visual Warehouse.
- Determining the type of data to be accessed.
- Rewriting the SQL query into the native file or database access language needed. A single SQL access could translate into multiple native data requests.
- Optimizing queries based on generic SQL query rewrite and file- or database-specific optimization.
- Querying multiple data sources for JOINs.
- Translating result sets into a consistent relational format, which involves restructuring nonrelational data into columns and rows.
- Sorting result sets as needed; for example, ORDER BY.
- Issuing all client catalog queries to the Classic Connect meta data catalog.

A Classic Connect data server accepts connection requests from the CROSS ACCESS ODBC driver and the sample application on MVS.

There are five types of services that can run in the data server:

- Region controller services, which include an MTO operator interface
- Initialization services
- Connection handler services

- Query processor services
- Logger services

Region controller services

The primary component of a data server is the region controller. The region controller is responsible for starting, stopping, and monitoring all of the other components of the data server. These different components are referred to as services. The services are implemented as individual load modules running as separate MVS tasks within the data server address space. Services can have multiple instances, and each instance can support multiple users.

The region controller determines which services to start based on SERVICE INFO ENTRY parameter settings.

Included in the region controller service is the MVS Master Terminal Operator (MTO) interface, which allows you to display and control the services and users that are being serviced by a data server. Using the MTO interface, you can also dynamically configure the data server.

Initialization services

Initialization services are special tasks that are used to initialize and terminate different types of interfaces to underlying database management systems or MVS system components. Currently, three initialization services are provided:

IMS BMP/DBB initialization service
used to initialize the IMS region controller to access IMS data using a BMP / DBB interface

IMS DRA initialization service
used to initialize the Classic Connect DRA interface and to connect to an IMS DBCTL region to access IMS data using the DRA interface

WLM initialization service
used to initialize and register with the MVS Workload Manager subsystem (using the WLM System Exit). This allows individual queries to be processed in WLM goal mode.

Connection handler services

A connection handler (CH) service task is responsible for listening for connection requests from Visual Warehouse. Connection requests are routed to the appropriate query processor task for subsequent processing.

Classic Connect supplies three typical transport layer modules that can be loaded by the CH task:

- TCP/IP
- SNA LU 6.2
- MVS cross memory services.

The MVS client application, DJXSAMP, can connect to a data server using any of these methods; however, the recommended approach for local clients is to use MVS cross memory services. Visual Warehouse can use either TCP/IP or SNA to communicate with a remote data server.

Query processor services

The query processor is the component of the data server that is responsible for translating client SQL into database- and file-specific data access requests. The query processor treats IMS and VSAM data as a single data source and is capable of processing SQL statements that access either IMS, VSAM, or both. Multiple query processors can be used to separately control configuration parameters, such as those that affect tracing and governors, to meet the needs of individual applications.

The query processor can service SELECT statements. The query processor invokes one or more subsystem interfaces (SSIs) to access the target database or file system referenced in an SQL request. The following SSIs are supported:

IMS BMP/DBB interface
Allows IMS data to be accessed through an IMS region controller. The region controller is restricted to a single PSB for the data server, limiting the number of concurrent users a data server can handle.

IMS DRA interface
Allows IMS data to be accessed using the IMS DRA interface. The DRA interface supports multiple PSBs and is the only way to support a large number of users. This is the recommended interface.

VSAM interface
Allows access to VSAM ESDS, KSDS, or RRDS files. This interface also supports use of alternate indexes.

Classic Connect supplies a RUNSTATS utility program that is used to update population statistics for the logical tables and their associated indexes and keys. This information can be used by the query processor to optimize JOINs.

Logger service

A logger service is a task that is used for system monitoring and trouble shooting. A single logger task can be running within a data server. During normal operations, you will not need to be concerned with the logger service.

Enterprise server

The enterprise server is an optional component that you can use to manage a large number of concurrent users across multiple data sources. An enterprise server contains the same tasks that a data server uses, with the exception of the query processor and the initialization services.

Figure 2 shows how the enterprise server fits into a Classic Connect configuration:

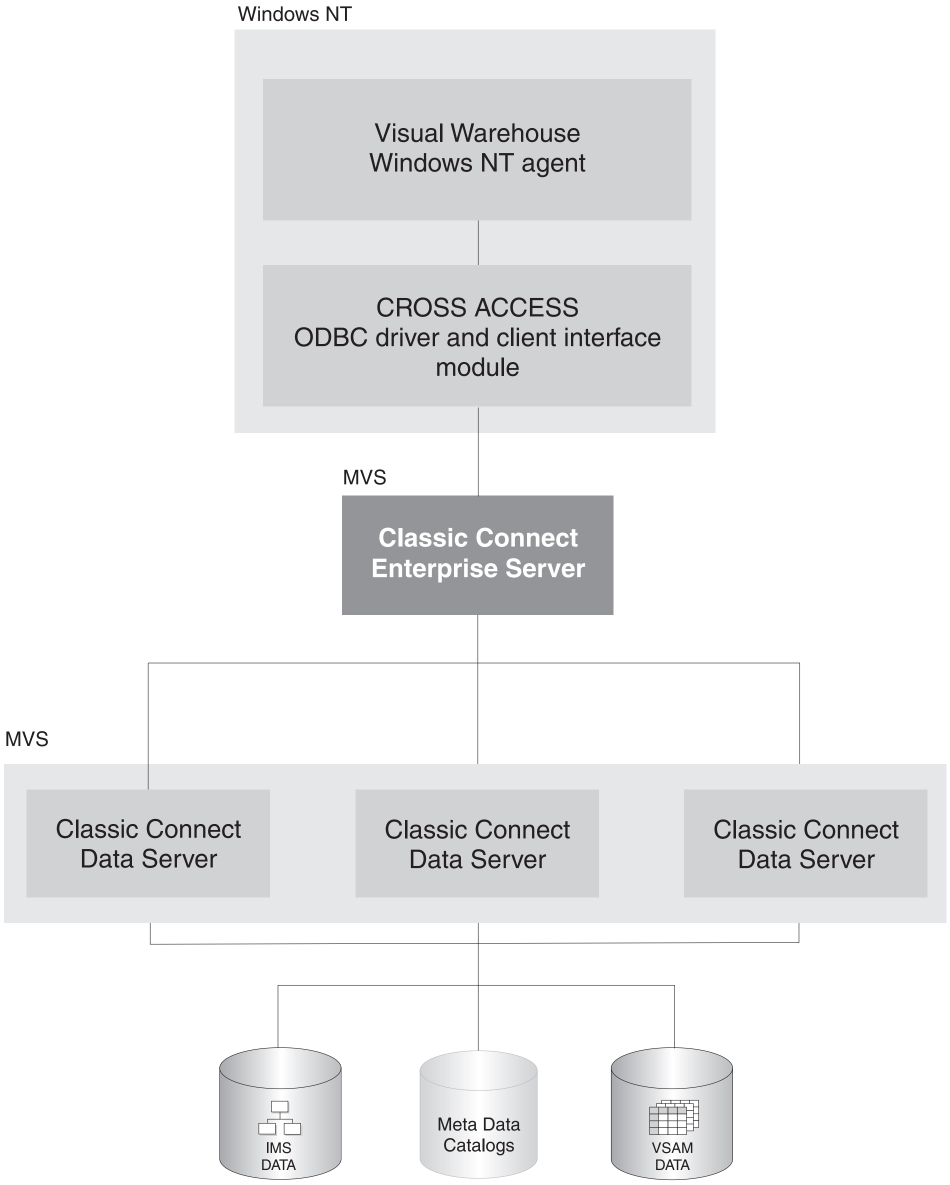

Windows NT
Visual Warehouse Windows NT agent
CROSS ACCESS ODBC driver and client interface module
MVS
Classic Connect Enterprise Server
MVS
Classic Connect Data Server
Classic Connect Data Server
Classic Connect Data Server
IMS DATA
Meta Data Catalogs
VSAM DATA

Figure 2. Classic Connect architecture with the enterprise server implemented

Like a data server, the enterprise server’s connection handler is responsible for listening for client connection requests. However, when a connection request is received, the enterprise server does not forward the request to a query processor task for processing. Instead, the connection request is forwarded to a data source handler (DSH) and then to a data server for processing. The enterprise server maintains the end-to-end connection between the client application and the target data server. It is responsible for sending and receiving messages between the client application and the data server.

The enterprise server is also used to perform load balancing. Using configuration parameters, the enterprise server determines the locations of the data servers that it will be communicating with and whether those data servers are running on the same platform as the enterprise server.

The enterprise server can automatically start a local data server if there are no instances active. It can also start additional instances of a local data server when the currently active instances have reached the maximum number of concurrent users they can service, or the currently active instances are all busy.

Data mapper

The Classic Connect nonrelational data mapper is a Microsoft® Windows®-based application that automates many of the tasks required to create logical table definitions for nonrelational data structures. The objective is to view a single file or portion of a file as one or more relational tables. The mapping must be accomplished while maintaining the structural integrity of the underlying database or file.

The data mapper interprets existing physical data definitions that define both the content and the structure of nonrelational data. The tool is designed to minimize administrative work, using a definition-by-default approach.

The data mapper accomplishes the creation of logical table definitions for nonrelational data structures by creating *meta data grammar* from existing nonrelational data definitions (COBOL copybooks). The meta data grammar is used as input to the Classic Connect meta data utility to create a meta data catalog that defines how the nonrelational data structure is mapped to an equivalent logical table. The meta data catalogs are used by query processor tasks to facilitate both the access and translation of the data from the nonrelational data structure into relational result sets.

The data mapper import utilities create initial logical tables from COBOL copybooks. A visual point-and-click environment is used to refine these initial logical tables to match site- and user-specific requirements. You can utilize the initial table definitions automatically created by data mapper, or customize those definitions as needed.

Multiple logical tables can be created that map to a single physical file or database. For example, a site may choose to create multiple table definitions that all map to an employee VSAM file. One table is used by department managers who need access to information about the employees in their departments; another table is used by HR managers who have access to all employee information; another table is used by HR clerks who have access to information that is not considered confidential; and still another table is used by the employees themselves who can query information about their own benefits structure. Customizing these table definitions to the needs of the user is not only beneficial to the end-user but recommended.

Figure 3 shows the data administration workflow with data mapper.

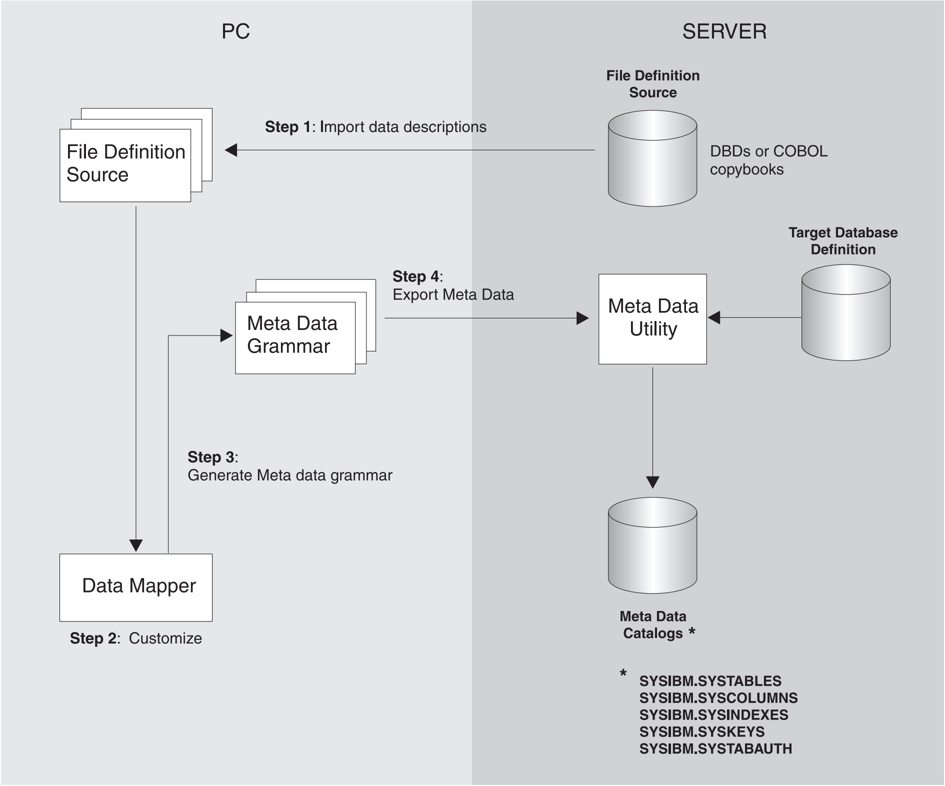


Figure 3. Data mapper workflow

The data mapper contains embedded FTP support to facilitate file transfer from and to the mainframe.

The steps in Figure 3 are described as follows:

1. Import existing descriptions of your nonrelational data into data mapper. COBOL copybooks and IMS database definitions (DBDs) can all be imported into the data mapper.

The data mapper creates default logical table definitions from the COBOL copybook information. If these default table definitions are acceptable, skip the following step and go directly to step 3.

2. Refine or customize the default table definitions as needed by the users. For example, importing the record layout for the VSAM customer master file creates the default Customer_Table. Two additional tables can also be created from the original:
   - Marketing_Customer_Table, which contains only those data items required by the marketing department
   - Service_Customer_Table, which contains only those data items required by support representatives
3. Generate the actual meta data grammar that will be used by the meta data utility on MVS.
4. Export the logical table definitions to the mainframe where the database or file resides. These definitions are then used as input to the meta data utilities, which create the meta data catalogs.

After completing these steps, you are ready to use Classic Connect operational components with your tools and applications to access your nonrelational data.

Chapter 2. Setting up the environment

This chapter summarizes the requirements for setting up integration between Classic Connect and Visual Warehouse.

Hardware and software requirements

The integration requires the following software:

- DataJoiner Classic Connect Version 2 Release 1
- Visual Warehouse Version 5 Release 2 CSD 2

Optionally, you can the DataJoiner Classic Connect data mapper to generate meta data grammar for you. You can obtain the data mapper from the following Web site:

```
http://www.software.ibm.com/data/datajoiner/news.html#newcxa
```

Installing and configuring prerequisite products

Complete the tasks summarized in Table 1 to set up integration between Classic Connect and Visual Warehouse. See the documentation listed in each task for more information.

Table 1. Summary of installation and configuration tasks

Task	Content	Location
Learning about the integration	What is Classic Connect?	"Chapter 1. What is Classic Connect?" on page 1
	Concepts and terminology	Chapter 2 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*

Table 1. Summary of installation and configuration tasks (continued)

Installing and configuring the data server	System requirements and planning	Chapter 3 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*
	Installing Classic Connect on MVS	Chapter 4 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*
	Data server installation and verification procedure	Chapter 6 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*
	Introduction to data server setup	Chapter 6 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*
	Configuring communication protocols between MVS and Windows NT	“Chapter 3. Configuring communications protocols between MVS and Windows NT®” on page 17
Installing and configuring the client workstation	Configuring a Windows NT client	“Chapter 4. Configuring a Windows NT client” on page 27
	Defining an agent site	Chapter 3 of *Managing Visual Warehouse* and the Visual Warehouse online help
Using an IMS or VSAM information resource	Mapping nonrelational data and creating queries	Chapter 13 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide* and *DataJoiner Classic Connect: Data Mapper Installation and User’s Guide*
	Optimization	Chapter 14 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*
	Defining an information resource	Chapter 3 of *Managing Visual Warehouse* and the Visual Warehouse online help

Table 1. Summary of installation and configuration tasks (continued)

Migrating from the Visual Warehouse Host Adapters	Migrating from the Visual Warehouse Host Adapters to Classic Connect	“Appendix. Migrating from the Visual Warehouse Host Adapters to Classic Connect” on page 39

Chapter 3. Configuring communications protocols between MVS and Windows NT®

Classic Connect supports the TCP/IP and SNA LU 6.2 (APPC) communication protocols to establish communication between a Visual Warehouse agent and Classic Connect data servers. A third protocol, cross memory, is used for local client communication on MVS.

This chapter describes modifications you need to make to the TCP/IP and SNA communications protocols before you can configure Classic Connect and includes the following sections:

- Communications options
- Configuring the TCP/IP communications protocol
- Configuring the LU 6.2 communications protocol

Communications options

Classic Connect supports the following communications options:

- Cross memory
- SNA
- TCP/IP

Cross memory

Cross memory should be used to configure the local MVS client application (DJXSAMP) to access a data server. Unlike for SNA and TCP/IP, there are no setup requirements to use the MVS cross memory interface. This interface uses MVS data spaces and MVS token naming services for communications between client applications and data servers.

Each cross memory data space can support up to 400 concurrent users, although in practice this number may be lower due to resource limitations. To support more than 400 users on a data server, configure multiple connection handler services, each with a different data space name.

The following example illustrates the communications compound address field:

```
XM1/DataSpace/Queue
```

Because you don’t need to modify any cross memory configuration settings, this protocol is not discussed in detail here.

Configuring communications protocols between MVS and Windows NT

SNA

SNA is a more sophisticated protocol and supports hardware compression, which can greatly reduce the amount of data actually transmitted over the wire. Unfortunately, the infrastructure requirements and setup time for using SNA are generally more expensive than using TCP/IP.

TCP/IP

A single TCP/IP connection handler service can service a maximum of 255 concurrent users. Depending on your particular TCP/IP subsystem, further limitations might apply.

Multiple sessions are created on the specified port number. The number of sessions carried over the port is the number of concurrent users to be supported plus one for the listen session the connection handler uses to accept connections from remote clients. If the TCP/IP implementation you are using requires you to specify the number of sessions that can be carried over a single port, you must ensure that the proper number of sessions have been defined. Failure to do so will result in a connection failure when a client application attempts to connect to the data server.

Configuring the TCP/IP communications protocol

This section describes the steps you must perform, both on your MVS system and on your Windows NT system, to configure a TCP/IP communications interface for Classic Connect. This section also includes a TCP/IP planning template and worksheet that are designed to illustrate TCP/IP parameter relationships.

There are two types of TCP/IP CIs that function with Classic Connect: IBM's TCP/IP and Berkeley Sockets. Your configuration may differ depending on which type of TCP/IP CI you are using. If your site does not use TCP/IP, proceed to "Configuring the LU 6.2 communications protocol" on page 22.

Both interfaces allow Classic Connect to communicate with the MVS TCP/IP stack. Berkeley Sockets allows you to use the host and service name, where IBM's TCP/IP requires a numeric IP address and port number. Berkeley Sockets can use a local host file instead of calling DNS. However, both Berkeley Sockets and IBM's TCP/IP require a TCP/IP address space name.

Configuring TCP/IP on MVS

Classic Connect's TCP/IP is compatible with both IBM's and Interlink's Berkeley Socket TCP/IP. This section describes how to configure Classic

Connect using IBM's TCP/IP. For more detailed information about IBM's or Interlink's TCP/IP, refer to the appropriate product's documentation.

Berkeley sockets is supported by IBM and Interlink. The Berkeley Sockets version requires an additional parameter in the DJXDSCF member called TASK PARAMETER, which identifies the Interlink subsystem name and identifies the location of the configuration data sets for IBM. Within the configuration data sets, users must specify the name of the started-task procedure used to start the TCP/IP address space name and can also specify the TCP/IP DNS IP addresses. If no environment variables are passed, then the default value TCPIP is used for both the address space name and as the high-level qualifier (hlq) of the standard configuration files:

- hlq.TCPIP.DATA
- hlq.ETC.HOSTS
- hlq.ETC.PROTOCOLS
- hlq.ETC.SERVICES
- hlq.ETC.RESOLV.CONF

Classic Connect uses a search order to locate the data sets, regardless of whether Classic Connect sets the hlq or not.

Determine the following values for the MVS system on which Classic Connect is being installed, and enter these values in the worksheet portion of Figure 4 on page 22.

IP address or hostname

Obtain either the hostname or IP address of the MVS system.

Using a hostname requires the availability of a configured local HOSTS file or a domain name server. If a domain name server is involved, then there is some overhead required to resolve the HOST name to the correct IP address. However, it is recommended that you use hostnames in remote client configuration files for readability and ease of future configuration changes.

Using hostnames also makes it easier to change IP addresses if the environment changes. If hostnames are used, frequently the data server/remote clients will not have to be reconfigured. Classic Connect can be brought down, and the network administrator can change the IP address for a hostname in the MVS and client domain name server(s). When the data server is restarted it will automatically listen on the new IP address for connection requests from remote clients. When a remote client connects to the data server it will automatically use the new IP address that has been assigned to the hostname without a change to the Classic Connect configuration files.

For IBM's TCP/IP, determine the IP address or hostname of the host computer on which Classic Connect is being installed. If you are running your MVS TCP/IP on off-loaded devices, specify the address of the TCP/IP stack on the MVS image, not the address of an off-load gateway's IP stack.

Port number

Obtain a unique port (socket) number greater than 1024 for each data server that will be accessed from a client.

The port number cannot be the same as any port that is already defined for use by any other application, including other Classic Connect data servers on the same MVS system. Using a non-unique port number causes the data server to fail at start up. To determine if a port number has already been assigned to another application, issue the following command from the Spool Display and Search Facility (SDSF) log:

```
TSO NETSTAT SOCKETS
```

Because some sites restrict the use of certain port numbers to specific applications, you should also contact your network administrator to determine if the port number you've selected is unique and valid.

Optionally, you can substitute the service name assigned to the port number defined to your system.

Service names, addresses, and tuning values for IBM's TCP/IP are contained in a series of data sets:

- hlq.TCPIP.DATA
- hlq.ETC.HOSTS
- hlq.ETC.PROTOCOLS
- hlq.ETC.SERVICES
- hlq.ETC.RESOLV.CONF

where ″hlq″ represents the high-level qualifier of these data sets. You can either accept the default high-level qualifier, TCPIP, or you can define a high-level qualifier specifically for Classic Connect.

When you have determined these values, use Figure 4 on page 22 to complete the MVS configuration of your TCP/IP communications.

Configuring TCP/IP on Windows NT

You must configure your Windows NT machine to locate the data server on MVS.

1. Resolve the host address on the client.

   If you are using the IP address in the client configuration file, you can skip this step.

   The client workstation must know the address of the host server to which it is attempting to connect. There are two ways to resolve the address of the host:

   - By using a name server on your network. This is the recommended approach. See your TCP/IP documentation for information about configuring TCP/IP to use a name server.

     If you are already using a name server on your network, proceed to step 2.

   - By specifying the host address in the local HOSTS file. On a Windows NT client, the HOSTS file is located in the %SYSTEMROOT%\SYSTEM32\DRIVERS\ETC directory.

     Add an entry to the HOSTS file on the client for the server's hostname as follows:

     ```
     9.112.46.200 stplex4a # host address for Classic Connect
     ```

     where 9.112.46.200 is the IP address, and stplex4a is the HOSTNAME. If the server resides in the same Internet domain as the client, this name can be a flat hostname. If the server is not in the same domain, the name must be a fully-specified domain name, such as stplex4a.stl.ibm.com, where stl.ibm.com is an example of a domain name.

   Notes:

   a. You must end the last line with a comment (# comment) or press Enter at the end of that line to insert an end-of-line character.

   b. You should refer to the documentation from your TCP/IP product for specific information about resolving host addresses.

2. Update the SERVICES file on the client.

   If you are using the port number in the client configuration file, you can skip this step.

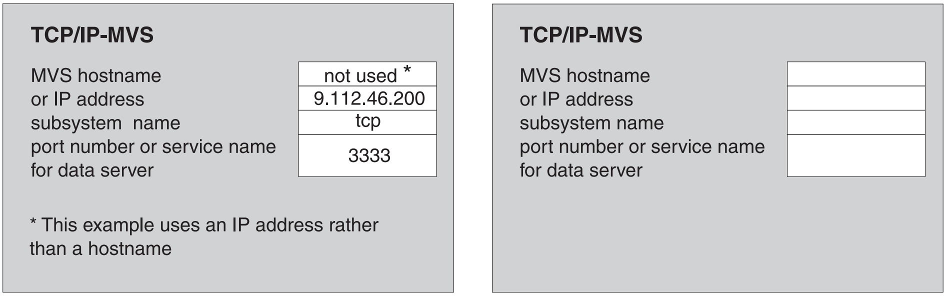
   The following information must be added to the SERVICES file on the client for TCP/IP support:

   ```
   ccdatser   3333   # CC data server on stplex4a
   ```

   The SERVICES file is located in the %SYSTEMROOT%\SYSTEM32\DRIVERS\ETC directory.

TCP/IP communications template and worksheet

The left side of Figure 4 on page 22 provides you with an example set of TCP/IP values for an MVS configuration; these values will be used during data server and client configuration in a later step. Use the right side of the figure as a template in which to enter your own values.



TCP/IP-MVS

MVS hostname	not used *
or IP address	9.112.46.200
subsystem name	tcp
port number or service name for data server	3333

* This example uses an IP address rather than a hostname

TCP/IP-MVS

MVS hostname	
or IP address	
subsystem name	
port number or service name for data server	

Figure 4. TCP/IP Communications Template and Worksheet

Configuring the LU 6.2 communications protocol

This section describes the values you must determine and steps you must perform, both on your MVS system and on your Windows NT system, to configure LU 6.2 (SNA/APPC) communications for Classic Connect.

Requirement:

For APPC connectivity between Classic Connect and DataJoiner for Windows NT, you need Microsoft SNA Server Version 3.0 with service pack 3 or later.

The information in this section is specific to Microsoft SNA Server Version 3.0. For more information about configuring Microsoft SNA Server profiles, see the appropriate product documentation. This section also includes a communications template and worksheet that are designed to clarify LU 6.2 parameter relationships on MVS and Windows NT and assist you with your LU 6.2 configuration.

Configuring LU 6.2 on MVS

If you will be using LU 6.2 to access Classic Connect from DataJoiner, you need to configure VTAM® table definitions on your MVS system, which include:

- A Mode table entry
- An Application ID

  The Application ID must be unique for a data server. Using a non-unique value causes the data server to fail at start up.

Unlike TCP/IP, you can specify the packet size for data traveling through the transport layer of an SNA network. However, this decision should be made by network administrators because it involves the consideration of complex routes and machine/node capabilities. In general, the larger the bandwidth of the communications media, or pipe, the larger the RU size should be.

Configuring LU 6.2 on Windows NT

This section explains the values you need to configure to use the SNA LU 6.2 protocol with your Windows NT client.

For each Windows NT system, configure the following values:

- SNA Server Properties Profile
- Connections for the SNA Server

  This example assumes you have installed a SNA DLC 802.2 link service. See your network administrator to obtain local and remote node information.
- Local APPC LU Profile

  The LU name and network must match the local node values in the Connections for the SNA Server profile. The LU must be an *Independent* LU type.
- Remote APPC LU Profile

  There must be one of these profiles for each Classic Connect data server or enterprise server that will be accessed. It must support parallel sessions. It will match the Application ID configured in VTAM table definitions on MVS discussed in “Configuring LU 6.2 on MVS” on page 22.
- APPC Mode

  See the mode, CX62R4K in Figure 5 on page 24. It has a max RU size of 4096.
- CPIC Symbolic Name

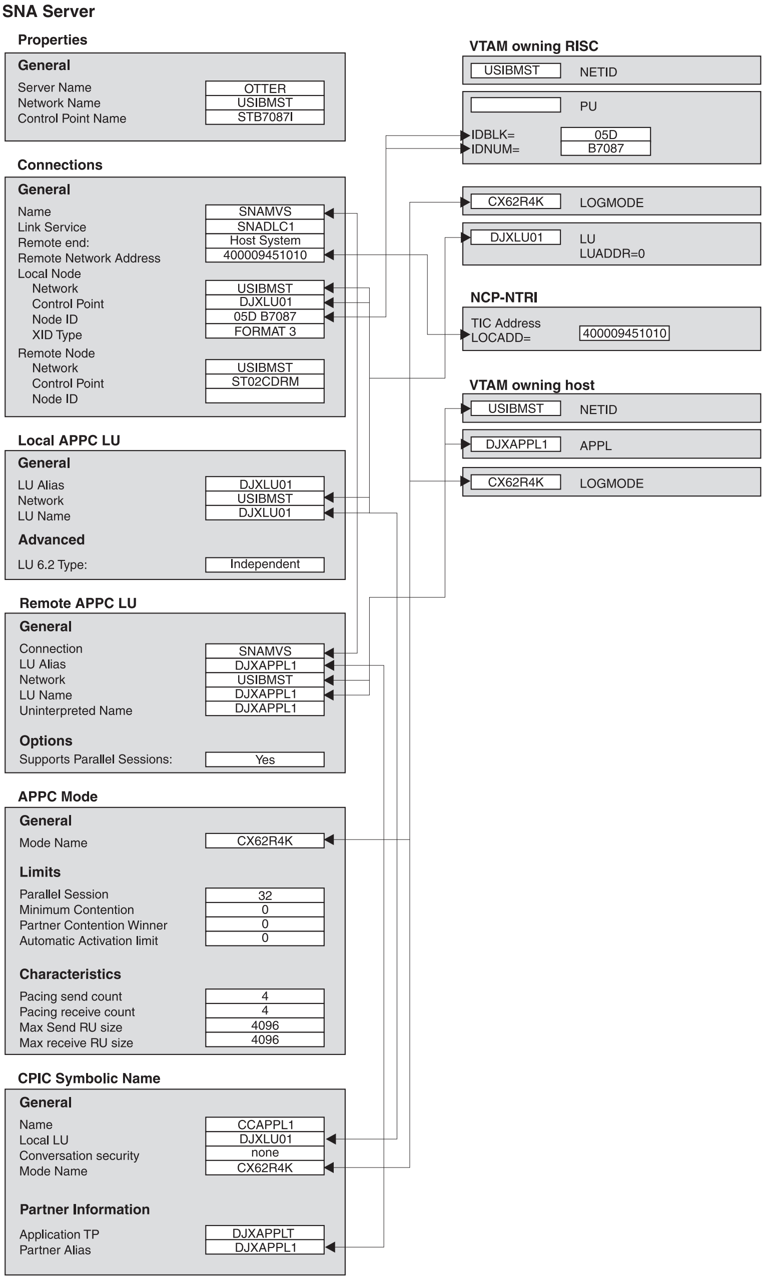
  There must be one of these for each Classic Connect data server or enterprise server that will be accessed. The TP Name referenced in this profile must be unique to the Classic Connect data server or enterprise server on a given MVS system.

After you have entered these values, save the configuration, and stop and restart your SNA Server. When the SNA Server and the Connection (in this example, OTTER and SNAMVS respectively), are ’Active,’ the connection is ready to test with an application.

APPC communications template and worksheet

Figure 5 on page 24, which provides you with an example set of VTAM and SNA values, is included for reference. Use Figure 6 on page 25, which is a duplicate of Figure 5 without the supplied values, as a worksheet in which you can enter the VTAM and SNA values specific to your LU 6.2

configuration. You will need the values you enter in this worksheet to complete configuration steps in subsequent chapters.



SNA Server

Properties

General

Server Name	OTTER
Network Name	USIBMST
Control Point Name	STB7087I

Connections

General

Name	SNAMVS
Link Service	SNADLC1
Remote end:	Host System
Remote Network Address	400009451010
Local Node	
Network	USIBMST
Control Point	DJXLU01
Node ID	05D B7087
XID Type	FORMAT 3
Remote Node	
Network	USIBMST
Control Point	ST02CDRM
Node ID	

Local APPC LU

General

LU Alias	DJXLU01
Network	USIBMST
LU Name	DJXLU01

Advanced

LU 6.2 Type:	Independent

Remote APPC LU

General

Connection	SNAMVS
LU Alias	DJXAPPL1
Network	USIBMST
LU Name	DJXAPPL1
Uninterpreted Name	DJXAPPL1

Options

Supports Parallel Sessions:	Yes

APPC Mode

General

Mode Name	CX62R4K

Limits

Parallel Session	32
Minimum Contention	0
Partner Contention Winner	0
Automatic Activation limit	0

Characteristics

Pacing send count	4
Pacing receive count	4
Max Send RU size	4096
Max receive RU size	4096

CPIC Symbolic Name

General

Name	CCAPPL1
Local LU	DJXLU01
Conversation security	none
Mode Name	CX62R4K

Partner Information

Application TP	DJXAPPLT
Partner Alias	DJXAPPL1

VTAM owning RISC

USIBMST	NETID
	PU
IDBLK=	05D
IDNUM=	B7087
CX62R4K	LOGMODE
DJXLU01	LU LUADDR=0

NCP-NTRI

TIC Address LOCADD=	400009451010

VTAM owning host

USIBMST	NETID
DJXAPPL1	APPL
CX62R4K	LOGMODE

Figure 5. LU 6.2 Configuration Template

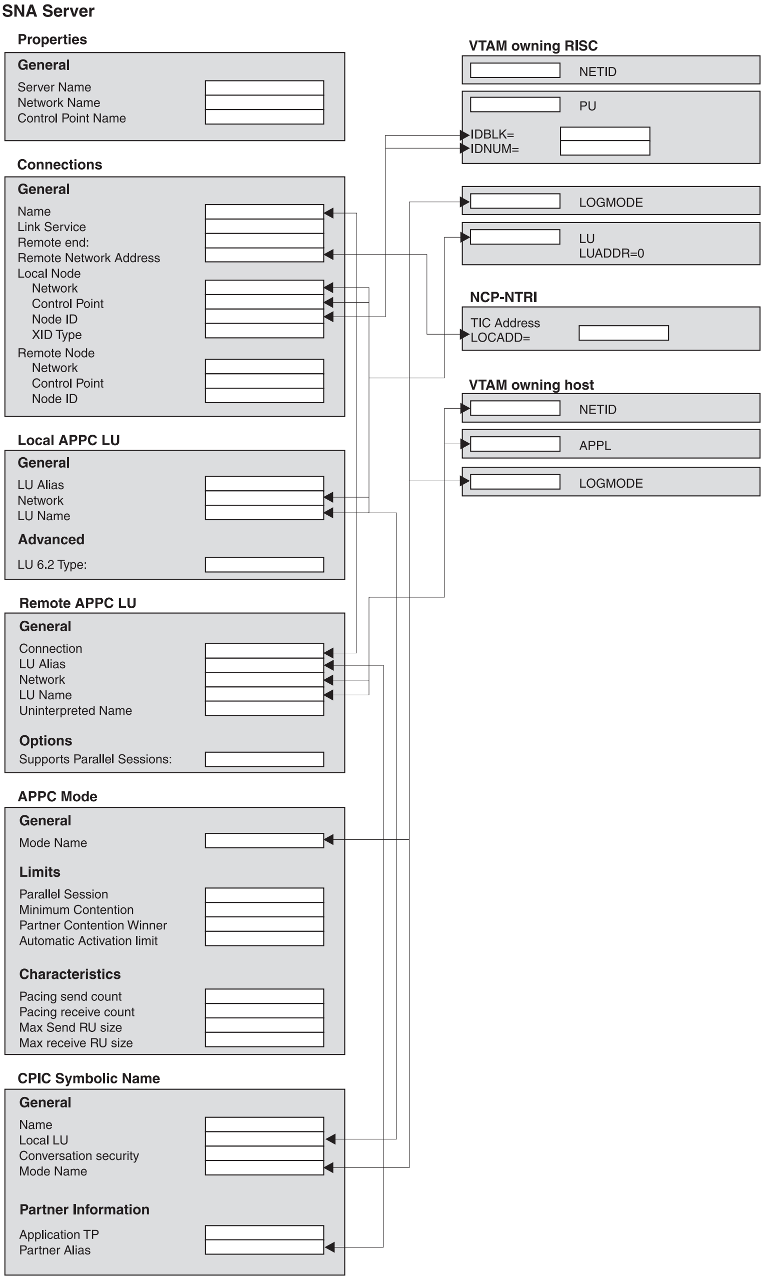


Figure 6. LU 6.2 Configuration Worksheet

Chapter 4. Configuring a Windows NT client

This chapter describes how to install the CROSS ACCESS ODBC driver, and use the driver to configure data sources.

Installing the CROSS ACCESS ODBC driver

The CROSS ACCESS ODBC driver is installed automatically with Visual Warehouse Version 5 Release 2 CSD 2. Select one of the following configurations:

- Stand-alone configuration
- Server configuration
- Windows NT agent configuration

Configuring data sources

CROSS ACCESS ODBC data sources are registered and configured using the ODBC Administrator. Configuration parameters unique to each data source are maintained through this utility.

You can define many data sources on a single system. For example, a single IMS system can have a data source called MARKETING_INFO and a data source called CUSTOMER_INFO. Each data source name should provide a unique description of the data.

Configuration prerequisites

The following information must be available before you attempt to configure the ODBC driver. If you are missing any of this information, see your system administrator.

- The name of the CROSS ACCESS data source to define in the ODBC Administrator
- If you are using TCP/IP:
  - The IP address for the host system where the data server runs.
  - The port number assigned to the TCP/IP connection handler in the TASK INFO ENTRY parameter of the data server.
- If you are using LU 6.2:
  - The name of the outbound side information record (SIR Outbound) defined in the SNA Server.
  - The data compression level of the host components.

Configuring a Windows NT client

Before you configure the ODBC driver, be sure that the Windows client is set up for the connection handler that you want to use, either TCP/IP or LU 6.2.

For APPC connectivity between Classic Connect and DataJoiner for Windows NT, you need Microsoft SNA Server Version 3 service pack 3 or later.

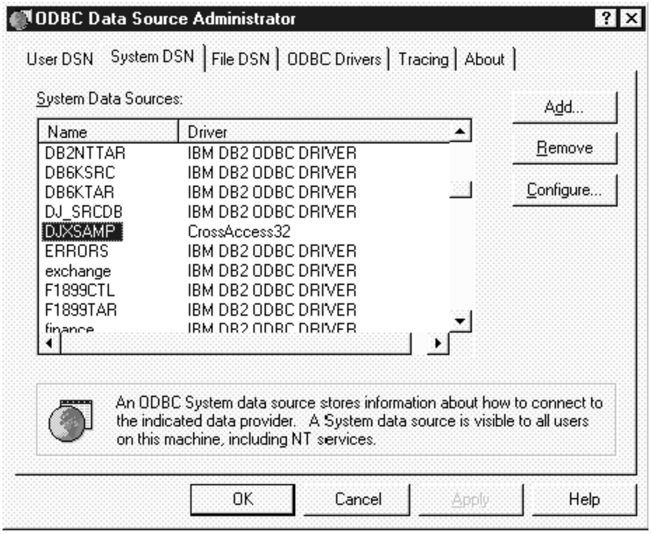
Specifying a data source

The data sources that are defined for all the currently installed ODBC drivers are listed in the ODBC Data Source Administrator window. From this window, you can:

- Add and configure data sources.
- Modify the configuration of data sources.
- Delete data sources.

To open the ODBC Data Source Administrator window:

1. Click **Start** on your desktop and click **Settings**.
2. Click **Control Panel**.
3. Open the ODBC icon. The ODBC Data Source Administrator window opens.



System Data Sources:

Name	Driver
DB2NTTAR	IBM DB2 ODBC DRIVER
DB6KSRC	IBM DB2 ODBC DRIVER
DB6KTAR	IBM DB2 ODBC DRIVER
DJ_SRCDB	IBM DB2 ODBC DRIVER
DJXSAMP	CrossAccess32
ERRORS	IBM DB2 ODBC DRIVER
exchange	IBM DB2 ODBC DRIVER
F1899CTL	IBM DB2 ODBC DRIVER
F1899TAR	IBM DB2 ODBC DRIVER
finance	IBM DB2 ODBC DRIVER

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

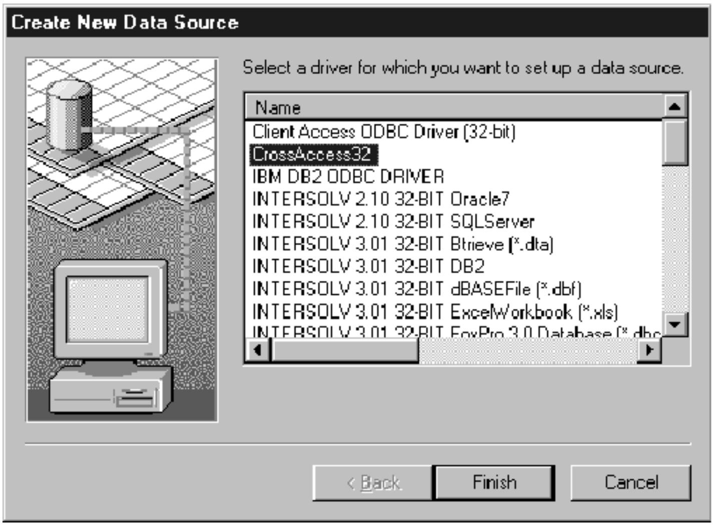
Figure 7. ODBC Data Source Administrator window

This window displays a list of data sources and drivers on the System DSN page.

Adding and configuring a data source

To add and configure a data source:

1. Open the ODBC Data Source Administrator window.
2. On the System DSN page, click **Add**. The Create New Data Source window opens.

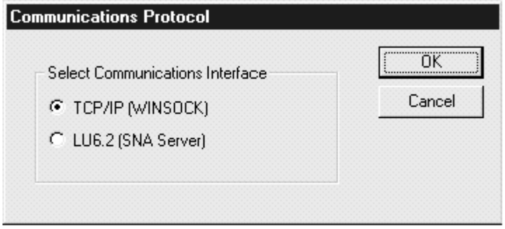


Select a driver for which you want to set up a data source.

Name
Client Access ODBC Driver (32-bit)
CrossAccess32
IBM DB2 ODBC DRIVER
INTERSOLV 2.10 32-BIT Oracle7
INTERSOLV 2.10 32-BIT SQLServer
INTERSOLV 3.01 32-BIT Btrieve (*.dta)
INTERSOLV 3.01 32-BIT DB2
INTERSOLV 3.01 32-BIT dBASEFile (*.dbf)
INTERSOLV 3.01 32-BIT ExcelWorkbook (*.xls)
INTERSOLV 3.01 32-BIT FoxPro 3.0 Database (*.dbc

Figure 8. Create New Data Source window

3. Click CrossAccess32.
4. Click **Finish**. The CROSS ACCESS Communications Protocol window opens.
5. Select a communications interface to use with the data source that you are configuring.



Select Communications Interface
● TCP/IP (WINSOCK)
○ LU6.2 (SNA Server)

Figure 9. Communications Protocol window

6. Click **OK**. The CROSS ACCESS ODBC Data Source Configuration window opens.

Configuring a Windows NT client

In this window, you can enter parameters for new data sources or modify parameters for existing data sources. Many of the parameters must match the values specified in the server configuration. If you do not know the settings for these parameters, contact the Classic Connect system administrator.

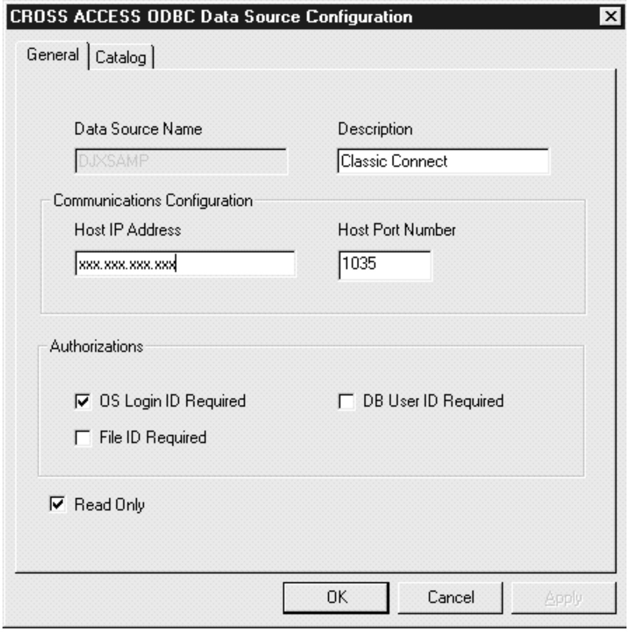
The parameters that you enter in this window vary depending on whether you are using the TCP/IP or LU 6.2 communications interface.

- For the TCP/IP communications interface, see “Configuring TCP/IP communications”.
- For the LU 6.2 communications interface, see “Configuring LU 6.2 communications” on page 32.

Configuring TCP/IP communications

Use the CROSS ACCESS ODBC Data Source Configuration window to:

- Name the data source.
- Configure the TCP/IP communications setting.
- Specify the necessary authorizations.



CROSS ACCESS ODBC Data Source Configuration

Data Source Name: DJXSAMP

Description: Classic Connect

Communications Configuration

Host IP Address: xxx.xxx.xxx.xxx

Host Port Number: 1035

Authorizations

☑ OS Login ID Required

□ DB User ID Required

□ File ID Required

☑ Read Only

Figure 10. The CROSS ACCESS ODBC Data Source Configuration window for TCP/IP

To configure TCP/IP communications:

1. Type the name of the data source in the **Data Source Name** field. This name must match Field 2 of the QUERY PROCESSOR SERVICE INFO ENTRY of the data server configuration file. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
2. Enter a brief description of the data source in the **Description** field.
3. Type the IP address of the data server in the **Host IP Address** field. This field specifies the IP address, in host-name or dotted-decimal notation, of the host where the data server is installed.
4. Type the port number (socket) assigned to the host component TCP/IP communications in the **Host Port Number** field. This number must match Field 10 of the TCP/IP SERVICE INFO ENTRY of the data server configuration file. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
5. Select one or more of the following check boxes:
   - **OS Login ID Required**. Select this box to be prompted for a user ID and password to log in to the operating system.
   - **DB User ID Required**. Select this box to be prompted for a user ID and password to log in to the database system, for example, DB2 or Sybase.
   - **File ID Required**. Select this box to be prompted for the file ID and password required to access the database. The file ID and password is required for certain databases, like Model 204.
6. Specify whether the data source has update capabilities. The default is read-only access.

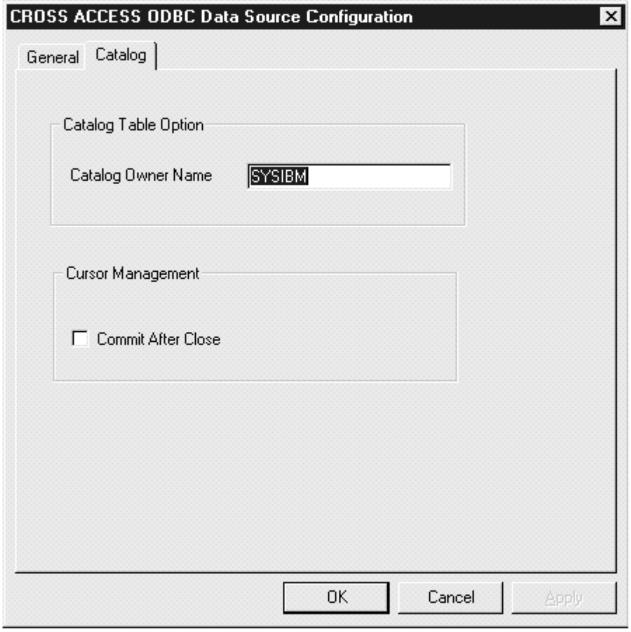
Setting the database catalog options

Use the Catalog page to perform the following tasks:

- Specify the catalog table option.
- Specify cursor management.

To set the database catalog options:

1. Click the **Catalog** tab in the CROSS ACCESS ODBC Data Source Configuration window.



Catalog Owner Name: SYSIBM

□ Commit After Close

Figure 11. CROSS ACCESS ODBC Data Source Configuration window

2. Type the name of the database catalog owner in the **Catalog Owner Name** field.
3. Select the **Commit After Close** check box if you want the ODBC driver to automatically issue a COMMIT call after a CLOSE CURSOR call is issued by the application. On certain database systems, a resource lock will occur for the duration of an open cursor. These locks can be released only by a COMMIT call and a CLOSE CURSOR call.

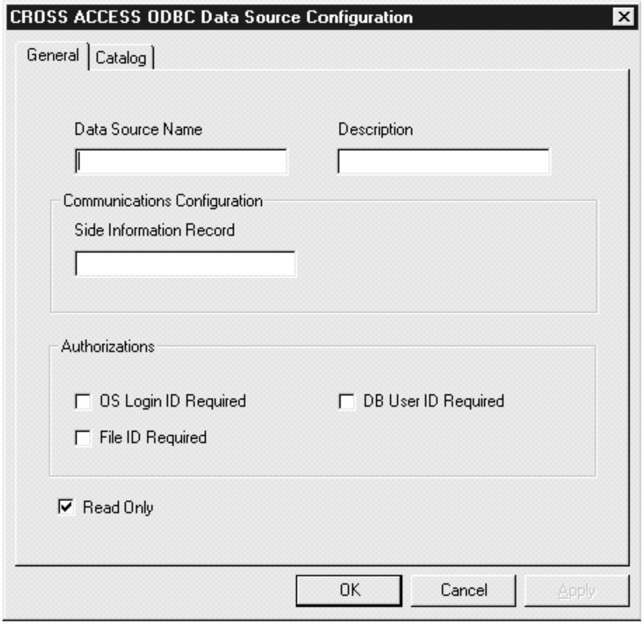
   If you leave this box clear, the cursors are freed without issuing a COMMIT call.
4. Click **OK**.

The TCP/IP communications information is saved.

Configuring LU 6.2 communications

Use the CROSS ACCESS ODBC Data Source Configuration window to:

- Identify the data source.
- Configure LU 6.2 communications settings.
- Specify necessary authorizations.



□ OS Login ID Required
□ DB User ID Required
□ File ID Required
☑ Read Only

Figure 12. The CROSS ACCESS ODBC Data Source Configuration window for LU 6.2

To configure LU 6.2 communications:

1. Type the name of the data source in the **Data Source Name** field. This name must match Field 2 of the QUERY PROCESSOR SERVICE INFO ENTRY of the data server configuration file. (For an example, see Chapter 9 of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*.)
2. Enter a brief description of the data source in the **Description** field.
3. Type the side information record (SIR) name in the **Side Information Record** field.

   The SIR name refers to a side information record (also called a CPIC symbolic name in Figure 6 on page 25) defined in the SNA server. This SIR must include the configuration parameters that represent the data server.
4. Select one or more of the following check boxes:
   - **OS Login ID Required**. Select this box to be prompted for a user ID and password to log in to the operating system.
   - **DB User ID Required**. Select this box to be prompted for a user ID and password to log in to the database system, for example, DB2 or Sybase.
   - **File ID Required**. Select this box to be prompted for the file ID and password required to access the database. The file ID and password is required for certain databases, like Model 204.

5. Clear the **Read Only** check box to indicate that the data source has update capabilities. The default is read-only access.

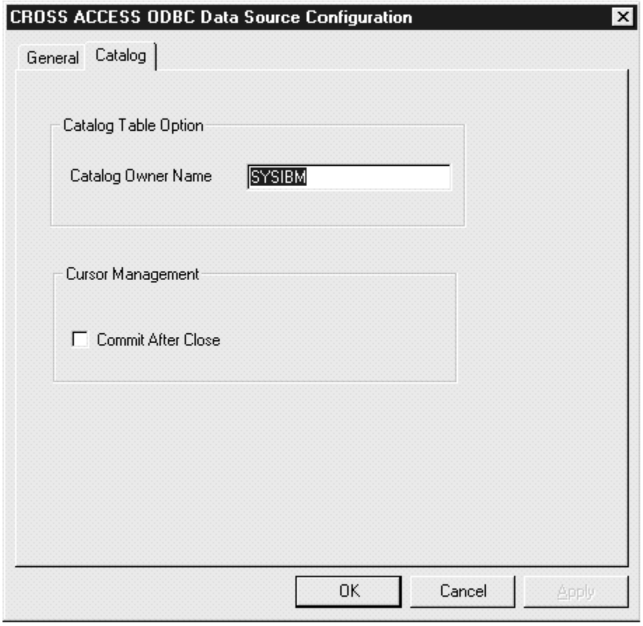
Setting the database catalog options

Use the Catalog page to perform the following tasks:

- Specify the catalog table option.
- Specify cursor management.

To set the database catalog options:

1. Click the **Catalog** tab in the Configuration window.



Catalog Owner Name: SYSIBM

□ Commit After Close

Figure 13. Database Catalog Options window

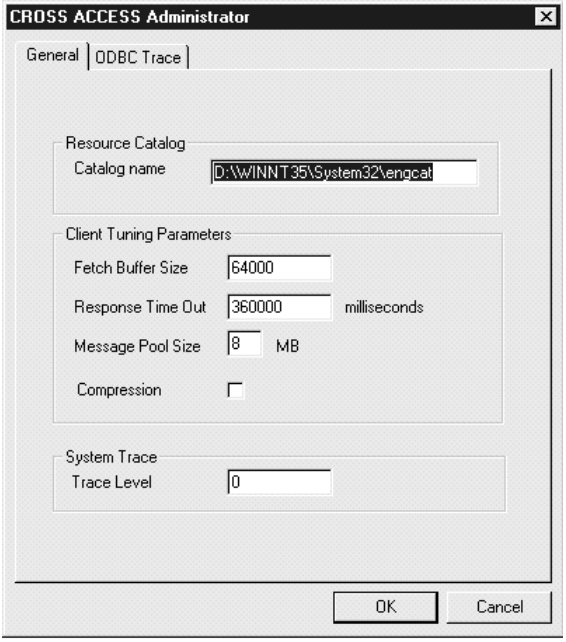
2. Type the name of the database catalog owner in the **Catalog Owner Name** field.
3. Select the **Commit After Close** check box if you want the ODBC driver to automatically issue a COMMIT call after a CLOSE CURSOR call is issued by the application. On certain database systems, a resource lock will occur for the duration of an open cursor. These locks can be released only by a COMMIT call and a CLOSE CURSOR call.

   If you leave this box clear, the cursors are freed without issuing a COMMIT call.
4. Click **OK**.

The LU 6.2 communications information is saved.

Configuring ODBC drivers

The CROSS ACCESS ODBC driver maintains a set of configuration parameters common to all CROSS ACCESS data sources. Configuration of these parameters are performed in the CROSS ACCESS Administrator window. The following steps show how to configure the ODBC driver parameters.



Catalog name: D:\WINNT35\System32\engcat

Fetch Buffer Size: 64000

Response Time Out: 360000 milliseconds

Message Pool Size: 8 MB

Compression: □

Trace Level: 0

Figure 14. General page of the CROSS ACCESS Administrator window

1. From the General page of the CROSS ACCESS Administrator window, type the full path name of the language catalog in the **Catalog name** field. This value is required.

   The language catalog contains messages in a specific language and is pointed to by a file contained within the CROSS ACCESS configuration files.

2. Optionally, type the size of a CROSS ACCESS fetch buffer in the **Fetch Buffer Size** field.

   This value tunes message blocking by controlling the amount of data that is returned in a single fetch request. The system packs as many rows of data as possible into a fetch buffer of the size specified. For example, if the fetch buffer is set to 10,000 bytes and each row is 2,000 bytes, the system can pack 5 rows per fetch request. If a single row is returned but does not fit into the specified buffer, the fetch buffer internally increases to fit the single row of data. To turn off message blocking, set this parameter to 1. The value must be from 1 to 64,000, inclusive. The default value is 10,000.

3. Optionally, type the maximum amount of time that this service will wait for an expected response before terminating a connection in the **Response Time Out** field.

   You can specify the following time intervals:

   - nMS = number of milliseconds
   - nS = number of seconds
   - nM = number of minutes

   Specify a value between 0 and 1000MS, 60S, or 60M. The default value is 6M.

4. Type the size of the memory used for all memory allocation in the **Message Pool Size** field. This value is required.

   Specify the number in bytes. The actual workable maximum value should be set to 2 MB less than the heap size. If the value specified is less than 1 MB, 1 MB is used. If the amount of storage that can be obtained is less than the value specified, the maximum amount available is obtained. The maximum permitted value is 2,097,152,000 bytes (2 GB). The default value is 1,048,575 bytes (1 GB).

5. Optionally, specify whether data compression is turned on or off for data transport between all tasks (internal and external) and between an initiator and the ODBC drivers. Type one of the following values in the **Compression** field:

   1 Data compression is turned on.

   2 Data compression is turned off.

   The default value is 2 (off).

6. Optionally, type a value that corresponds to the amount of information that the ODBC driver writes to the trace log in the **Trace Level** field. The value must be an integer between 0 and 4, where:

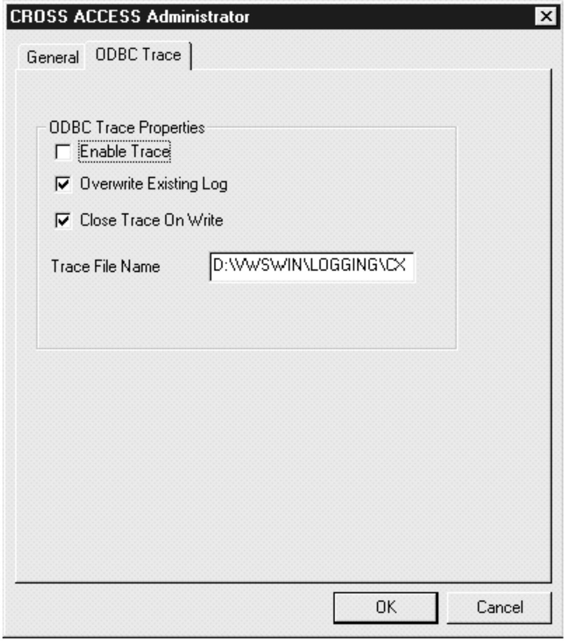
   0 No tracing information is logged.

   1 Minimum tracing information is logged.

   4 Maximum tracing information is logged.

   This trace is different from the ODBC trace; it is specific to the ODBC driver used by Visual Warehouse.

7. Click the **ODBC Trace**tab in the CROSS ACCESS Administrator window.



ODBC Trace Properties

□ Enable Trace

☑ Overwrite Existing Log

☑ Close Trace On Write

Trace File Name: D:\VWSWIN\LOGGING\CX

Figure 15. ODBC Trace page of the CROSS ACCESS Administrator window

8. Select the **Enable Trace** box to generate an ODBC trace.
9. Select the **Overwrite Existing Log** box to overwrite an existing log trace.
10. Select the **Close Trace on Write** box if you want the driver to close the trace log after each message is written.
11. Specify the name of the trace file in the **Trace File Name** field.

    If the directory is not indicated, the trace file will be created in the subdirectory of the Program Files directory that corresponds to the tool that issued the query to the ODBC data source.
12. Click **OK**.

Appendix. Migrating from the Visual Warehouse Host Adapters to Classic Connect

To migrate from the Visual Warehouse Host Adapters to Classic Connect:

1. Edit the Meta Data Utility. The sample Meta Data Utility JCL is found in the SDJXSAMP member DJXMETAU. See Appendix A of *DataJoiner Classic Connect: Installation, Configuration, and Reference Guide* for more information.
2. Supply a valid job card and modify the DJX high-level qualifier.
3. If you are mapping IMS data, modify the IMS high-level qualifier and uncomment the DBDLIB DD statement.
4. If you are referencing VSAM files by DD names, add DD statements with the same DD names in the Meta Data Utility JCL.
5. The first time you run the Meta Data Utility, you need to create the meta data catalog files. Unlike the Visual Warehouse Host Adapters, both IMS and VSAM meta data can exist in the same catalog.
6. Define the meta data grammar files that were used with the Visual Warehouse Host Adapters as input to the Meta Data Utility. As noted above, both IMS and VSAM meta data can exist in the same catalog.
7. Verify execute and access authority for the data sets referenced in the JCL.
8. Submit the Meta Data Utility JCL for execution and review the output.
9. If necessary, install and configure the CROSS ACCESS ODBC driver.
10. Adapt the Visual Warehouse Host Adapter setup to use Classic Connect:
    1. Locate the IMS and VSAM information resource on the Visual Warehouse Desktop **Sources** tab.
    2. Open the Information Resource notebook.
    3. Select the **Database** tab.
    4. Change the data source name to be the same as the data source name used when configuring the CROSS ACCESS ODBC driver.
    5. Close the Information Resource notebook.
11. Run a business view that uses the modified information resource to ensure it runs successfully.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue,
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Other data sources, including IDMS, M204, ADABAS, Datacom/DB and Sequential Files are available directly from CROSS ACCESS Corporation, Oakbrook Terrace, IL at http://www.crossaccess.com.

Trademarks

DataGuide
DataJoiner
DB2
IBM
IMS
MVS
Visual Warehouse
VTAM

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

To get copies of the books listed here, or to get more information about a particular library, see your IBM representative.

- *IBM DB2 DataJoiner Classic Connect: Installation, Configuration, and Reference Guide*, SC26-9319
- *IBM DB2 DataJoiner Classic Connect Data Mapper: Installation and User's Guide*, SC26-9318
- *IBM Visual Warehouse for Windows NT: Managing Visual Warehouse*, SC26-8822
- *IBM Visual Warehouse for Windows NT: Planning and Installing Visual Warehouse and DataGuide*®, SC26-3496

Index

Part Number: CT624NA
Program Number: 5639–VW5

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-9857-00

CT624NA