

DB2 DataJoiner[®] for Windows NT Systems



Planning, Installation, and Configuration Guide

Version 2 Release 1 Modification 1

DB2 DataJoiner[®] for Windows NT Systems



Planning, Installation, and Configuration Guide

Version 2 Release 1 Modification 1

Note

Before using this information and the product it supports, read the information in “Notices” on page 299.

Third Edition (October 1999)

This edition replaces SC26-9150-01.

© **Copyright International Business Machines Corporation 1995, 1999. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Welcome to DataJoiner	ix
Highlighting Conventions	ix
Graphical Conventions	x
How to Read the Syntax Diagrams	x

Part 1. Introduction to DataJoiner 1

Chapter 1. About DataJoiner	3
Working with Data Using DataJoiner	4
Accessing DataJoiner from Client Workstations	4
Database Administration Tools	6
The Command Line Processor	7
Database Director	7
Visual Explain.	7
DB2 Performance Monitor.	7
Application Development Using the Software Developer's Kit	8
Internet Connection Using Net.Data	9
Host Data Access from the Desktop Using DataJoiner DDCS	10
Operating System Security	13
Local and Domain Database Access	13
Remote Database Access	13
User ID Authorization Level Verification	14
Heterogeneous Data Replication	14
Differences in the DataJoiner for Windows NT Environment	15
Chapter 2. What's New in DataJoiner Version 2?	17

Part 2. Planning for and Installing DataJoiner 21

Chapter 3. Planning for Installation	23
Hardware Requirements	23
Memory and Fixed Disk Estimating Table	24
Software Requirements	25
Requirements for the DataJoiner Server	25
Requirements for DataJoiner Data Sources	26
Client Access to DataJoiner	27
Supported Compilers for DataJoiner	28
Replication Considerations	28

License Information	29
-------------------------------	----

Chapter 4. Installing DataJoiner. 31

Installation Notes	31
Creating a DataJoiner Administrator User Name	32
Installing from CD-ROM.	33
Verifying the Installation.	34
Verifying Instances.	34
Checking the Programs Menu	35
Creating the Sample Database and Selecting Sample Data.	35

Chapter 5. Migrating from Previous Versions 37

Migrating Instances	37
Migrating Databases	37
Correcting Migration Inhibitors	38
Selective Migration.	39

Chapter 6. Getting Started with DataJoiner 41

Logging on to the System	41
Working with DataJoiner Instances	41
Starting and Stopping DataJoiner	42
Starting and Stopping DataJoiner Using the Command Line	42
Starting and Stopping DataJoiner Using Windows NT Tools.	43
Creating a DataJoiner Database	43
Parameters	44
Running Your Own Applications	47
Running ODBC-Enabled Applications	47

Part 3. Connecting DataJoiner to Data Sources. 49

Chapter 7. Configuring Access to Data Sources 51

Loading Data Access Modules	52
Configuring for Two-Phase Commit	52
Enabling Data Sources for Two-Phase Commit	54
Configuring DataJoiner for Two-Phase Commit	55

Overview of Configuration Steps	56	Chapter 14. Accessing Sybase Data Sources Using Sybase Open Client	97
Chapter 8. Accessing DRDA Data Sources	57	Specifying a Connect String for Open Client	100
Accessing DRDA Data Sources Using Nicknames and Pass-through	57	Syntax	100
Step 1: Configure Network Communications	58	Examples	101
Step 2: Update the Node Directory	58	Chapter 15. Accessing Sybase SQL Anywhere Data Sources Using ODBC	103
Step 3: Update DataJoiner Catalog Views with Server and User Information.	59	Chapter 16. Accessing Teradata Data Sources	105
Accessing DRDA Data Sources Using DataJoiner DDCS	62	Default Type Mappings from Teradata Data Sources to DataJoiner	108
Step 1: Configure Network Communications	62	Default Type Mappings from DataJoiner to Teradata Data Sources	110
Step 2: Update the Node Directory	62	Chapter 17. Accessing IMS and VSAM Data Sources Using Classic Connect	111
Step 3: Catalog a DCS Directory Entry	63	Chapter 18. Accessing Data Sources Using a Generic Data Access Module	113
Step 4: Catalog a Database Directory Entry	64	Chapter 19. Configuring Communications for DRDA Data Sources	117
Step 5: Bind Packages Used by DataJoiner and Other Applications	64	Configuring DataJoiner as a DRDA Application Server	117
Chapter 9. Accessing DB2RA Data Sources	67	Configuring DataJoiner DDCS Using SNA	118
Accessing DB2RA Data Sources Using Nicknames and Pass-through	67	Step 1: Gathering Information About the DRDA Servers	118
Step 1: Configure Network Communications	67	Step 2: Gathering DDCS Information	119
Step 2: Update the Node Directory	73	Step 3: Gathering Information for Each Server Connection	120
Step 3: Update DataJoiner Catalog Views with Server and User Information.	74	Step 4: Configuring SNA LU 6.2	120
Accessing DB2RA Data Sources Outside the Heterogeneous Environment	76	Configuring DataJoiner DDCS Using TCP/IP	135
Step 1: Configure Network Communications	76	<hr/>	
Step 2: Update the Node Directory	77	Part 4. Using DataJoiner	139
Step 3: Catalog a Database Directory Entry	77	Chapter 20. Performing Administration Tasks	141
Step 4: Bind Packages used by DataJoiner and Other Applications	77	Managing Database Directories Using Database Director	141
Chapter 10. Accessing Informix Data Sources	81	Tuning SQL Statements Using Visual Explain	141
Using the Informix Version 7 Data Access Module	81	Tuning Performance Using the DB2 Performance Monitor	143
Chapter 11. Accessing Microsoft SQL Server Data Sources Using ODBC	85	Snapshot Monitoring.	143
Chapter 12. Accessing Oracle RDB Data Sources Using ODBC	89	Event Monitoring.	144
Chapter 13. Accessing Oracle Data Sources Using SQL*Net or Net8	91	Chapter 21. Setting Environment Variables	147

DataJoiner Environment Variables	148	NULL Values in Before-Image Columns	185
Performance	148	Subsetting Target Tables.	185
General	148	Vertical Subsetting	185
Command Line Processor	150	Horizontal Subsetting	186
Communications	151	Combining Data	186
ADSM	152	SQL Statements and Stored Procedures	187
Part 5. Replicating Data 155		Pull versus Push Apply Design	188
Chapter 22. Data Replication—What is it? 157		Update-Anywhere Replication	191
The IBM Data Replication Solution	157	Views for Sources and Targets	191
Support of Replication Environments	158	Source Views	191
Replication Solution and Product		Target Views	192
Environment	159	Preventing Gaps in the Target Table.	192
Multivendor Data Sources and Targets.	160	Coordination of Source Security and Target Security	192
Chapter 23. The Replication Tools 161		Promote Functions	192
The Replication Administration Tool	161	IBM Replication Solution Restrictions	193
Administering Replication: Overview	163	Chapter 25. Replication Requirements 197	
How the Replication Administration Tool Manages the Replication Environment	164	Supported Sources and Targets	197
DataJoiner	167	The Replication Administration Tool	
The Capture Program for DB2 Sources.	168	Hardware Requirements	198
Capture Triggers for Non-IBM Sources.	168	DB2 DPROP for Microsoft Jet Hardware Requirements	198
Capture Triggers Overview	168	The Replication Administration Tool	
How the Capture Triggers Capture the Data Changes	169	Software Requirements	198
Tables Used by the Capture Triggers	171	DB2 DPROP for Microsoft Jet Software Requirements	201
The Apply Program	172	Capture and Apply Program Requirements	202
The Apply Qualifier	173	Authorization Requirements	202
Control Tables Used by the Apply Program	174	Authorization Requirements for Administration.	203
The Apply Program and Control Servers	176	Authorization Requirements for the Capture Program	203
The Apply Processing Cycle	176	Authorization Requirements for the Apply Program	204
Chapter 24. Data Replication Concepts 179		System-Specific Privilege Levels	205
Subscription Members and Sets	179	Mapping of User IDs	206
Types of Copies	179	Chapter 26. Setting up the DataJoiner Environment for Replication 211	
Full-Refresh and Differential Refresh	180	Setting up DataJoiner for Windows NT	211
Changed Data Tables	181	Installing DataJoiner	211
CD and CCD Tables	181	Setting up an Instance	211
Pruning the CD and CCD Tables.	183	Configuring Database Connections	212
Transaction-Based versus Transaction-Consistent Replication: Using Internal CCD Tables to Reduce Network Load	183	Connecting Clients to DataJoiner.	213
Before-Image Columns and After-Image Columns.	184	What to Do Next	213
Before-Image Column Values	185	Setting up DB2 DPROP for Microsoft Jet	214
		Preparing the Server	214
		Preparing the Client	214

Chapter 27. Installing the Replication Administration Tool and Connecting All Databases	217
Installing the Replication Administration Tool	217
Configuring Access from the Replication Administration Tool to DataJoiner and DB2	219
Setting Administrative Preferences	219
Binding to Source, Target, and Control Servers	220
Creating Replication Control Tables	221
Binding the Capture and Apply Programs in DB2 Systems	222
Finalizing the Setup for the Apply Program in NT	223
Binding the Apply Program in NT	223
Providing End-User Authentication in NT	224
Setting up the NT Service Control Manager	225
Chapter 28. Starting and Using the Replication Administration Tool	227
Starting the Replication Administration Tool	227
General Steps for Setting up Replication	228
Editing the Replication Administration Tool Logic	229
Editing the Replication Administration Tool-Generated SQL	229
Running the Replication Administration Tool-Generated SQL	230
Running Capture and Apply	230
Operating Apply for UNIX platforms	230
Chapter 29. DataJoiner with the Replication Administration Tool: Data Typing	235
DB2 to Oracle Replication	236
DB2 to Informix	236
DB2 to Microsoft SQL Server, Sybase, or Sybase SQL Anywhere	237
DB2 to Microsoft Jet	238
Chapter 30. Non-IBM Databases: Table Names, Control Tables, and the CCD	241
Table Names and Qualifiers	241
Control Tables in a Non-IBM Database	241
CCD Table and Capture Triggers	241
CCD Nicknames in DataJoiner	242
The Register and Pruning Control Tables	242

Part 6. Appendixes 243

Appendix A. DRDA Application Server	245
Supported DRDA Functions	245
Bind Options Supported by the DataJoiner DRDA Application Server	245
Restrictions	250
Configuration Considerations	251
CCSID Support	251
Supported DRDA Application Requesters	252
MVS DRDA Application Requesters	252
DDCS DRDA Application Requesters	252
DB2 for VM DRDA Application Requesters	252
DB2 for AS/400 DRDA Application Requesters	253
Appendix B. Configuration Parameters	255
Appendix C. Worksheets for Use with DataJoiner DDCS	261
MVS, VSE, and VM Server Worksheet	261
OS/400 Server Worksheet	263
Windows NT Configuration Worksheet	264
Windows NT Connection to Server Worksheet	264
Appendix D. National Language Support	265
Conversion Between Different Code Pages	265
When Does Character Conversion Occur?	265
Deriving Code Page Values	266
Supported Character Conversions	268
Country Code and Code Page Support	271
Date and Time Formats	276
Appendix E. Diagnosing Installation Problems on Windows 95/Windows NT.	279
Handling Installation Errors	279
Appendix F. Resolving Problems Encountered by Applications That Predate Version 2.1.1	283
Querying System Catalog Tables and Views	283
Changes	284
Problems	286
Resolution	286
Modifying System Catalog Tables	287
Change	287
Problem	287

Resolution	288	DataJoiner Administration	296
Appendix G. Where to Learn More about DataJoiner, DB2 for CS, and Replication Products	289	DataJoiner Services	296
DataJoiner, DB2 for CS, and Replication Publications.	289	First Phase: Planning.	296
How to Order, View, and Print Publications	294	Second Phase: Implementation	297
Internet Resources	294	Notices	299
Appendix H. DataJoiner Classes and Services	295	Trademarks	301
DataJoiner Classes	295	Index	303
Using DataJoiner	295	Readers' Comments — We'd Like to Hear from You	313

Welcome to DataJoiner

This book provides you with planning information and installation and configuration procedures for DataJoiner for the Windows NT operating system. This book is for database administrators, system administrators, and any other knowledge workers who have responsibility for administering and operating in a heterogeneous environment.

Part 1 gives a brief overview of the functionality that is provided by DataJoiner and lists the new features for this version.

Part 2 provides information about planning for and installing DataJoiner, as well as instructions for performing fundamental administrative tasks.

Part 3 describes how to configure different data sources to communicate with DataJoiner.

Part 4 describes how to use DataJoiner and describes its environment variables and their uses.

Part 5 contains information about replication, including planning, configuring, defining, replication sources and targets, and maintaining replication objects.

Part 6 contains information about configuring DataJoiner as a DRDA application server, configuration parameters, configuration worksheets, and instructions for viewing, printing, and ordering additional information.

Highlighting Conventions

This book uses these highlighting conventions:

Boldface type

Indicates commands and graphical user interface (GUI) controls (for example, names of fields, names of folders, menu choices). Boldface type also indicates examples of SQL keywords in the *Application Programming and SQL Reference Supplement*.

Monospace type

Indicates examples of coding or of text that you type.

Italic type

Indicates variables that you should replace with a value. Italic type also indicates book titles and emphasizes words.

UPPERCASE TYPE

Indicates SQL keywords and names of objects (for example, tables, views, and servers).

Graphical Conventions

This book uses these graphical conventions:



This icon marks a fast path — fast paths tell you if you can skip sections to get to the next "how to" information. Fast path also indicate where you should go next.



This icon marks a tip — additional information that can help you complete a task.



This icon indicates that a TaskGuide is available to help you perform a task that you would otherwise have to perform manually.

How to Read the Syntax Diagrams

The following rules apply to the syntax diagrams used in this book:

Arrow symbols

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

- ▶— Indicates the beginning of a statement.
- ▶ Indicates that the statement syntax is continued on the next line.
- ▶— Indicates that a statement is continued from the previous line.
- ▶▶ Indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the ▶— symbol and end with the —▶ symbol.

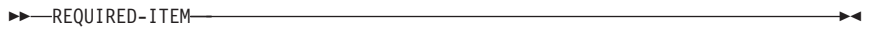
Conventions

- SQL commands appear in uppercase.
- Variables appear in italics (for example, *column-name*). They represent user-defined parameters or suboptions.
- When entering commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
- Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as given.
- Footnotes are shown by a number in parentheses, for example, (1).

- A `␣` symbol indicates one blank position.

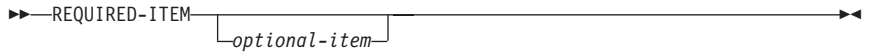
Required items

Required items appear on the horizontal line (the main path).



Optional items

Optional items appear below the main path.

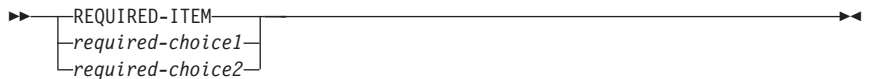


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.



Multiple required or optional items

If you can choose from two or more items, they appear vertically in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.

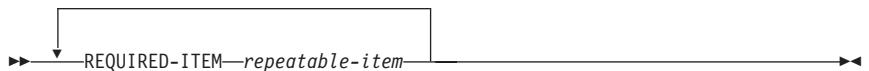


If choosing one of the items is optional, the entire stack appears below the main path.

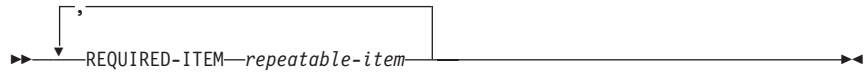


Repeatable items

An arrow returning to the left above the main line indicates that an item can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can specify more than one of the choices in the stack.

Default keywords

IBM-supplied default keywords appear above the main path, and the remaining choices are shown below the main path. In the parameter list following the syntax diagram, the default choices are underlined.



Part 1. Introduction to DataJoiner

Chapter 1. About DataJoiner

IBM's DataJoiner allows you to access data residing on multiple and diverse platforms, both IBM and multi-vendor, relational and non-relational, as a single database image. With DataJoiner, you can access all the data in your enterprise as if it were local.

DataJoiner V2.1.1 consists of DB2 Version 2 products and books, additional multi-database functionality, replication functionality, and other IBM products, including:

The DataJoiner server

A *relational* database management system containing functions and tools that you can use to create, update, control, and manage relational databases using Structured Query Language (SQL)

Software Developer's Kit (SDK) functionality

A collection of tools for database application developers

Net.Data

A Web server gateway that application developers can use to create Internet applications that access data from DB2 database

The replication administration tool

The administrative component of heterogeneous replication, which you can use to define and prepare relational data sources and targets for replication. The replication administration tool runs on a Windows 95 or Windows NT client

Apply program

The replication component that provides the actual replication functionality

Spatial Extender

The component that enables DataJoiner to store and manipulate geographic data types

Documentation

The entire DataJoiner library and relevant DB2 books in HTML and PostScript formats

In addition to these components:

- Remote clients need the **DB2 Client Application Enabler (CAE)**. With the DB2 CAE, client workstations from a variety of platforms can connect to any DataJoiner server. The DB2 CAE CD-ROM is included with DataJoiner.

- If a DB2 client is a replication source, the Capture program must be installed on the client.

Working with Data Using DataJoiner

DataJoiner includes graphical tools that you can use to tune performance, access remote DB2 and DataJoiner servers, manage all servers from a single site, develop powerful applications, and process SQL queries.

DataJoiner includes the DB2 database engine and the Administrator's Toolkit, which provide the following components:

- The **Command Line Processor**, used to access and manipulate databases from the system command prompt
- The **Database Director** tool, used for administrative tasks such as configuring the system, managing directories, backing up and recovering the system, and managing media
- The **DB2 Performance Monitor**, used to monitor the performance of your DataJoiner system for tuning purposes
- The **Visual Explain** tool, used to graphically view and navigate complex SQL access plans

See “Database Administration Tools” on page 6 for more information about these components.

Accessing DataJoiner from Client Workstations

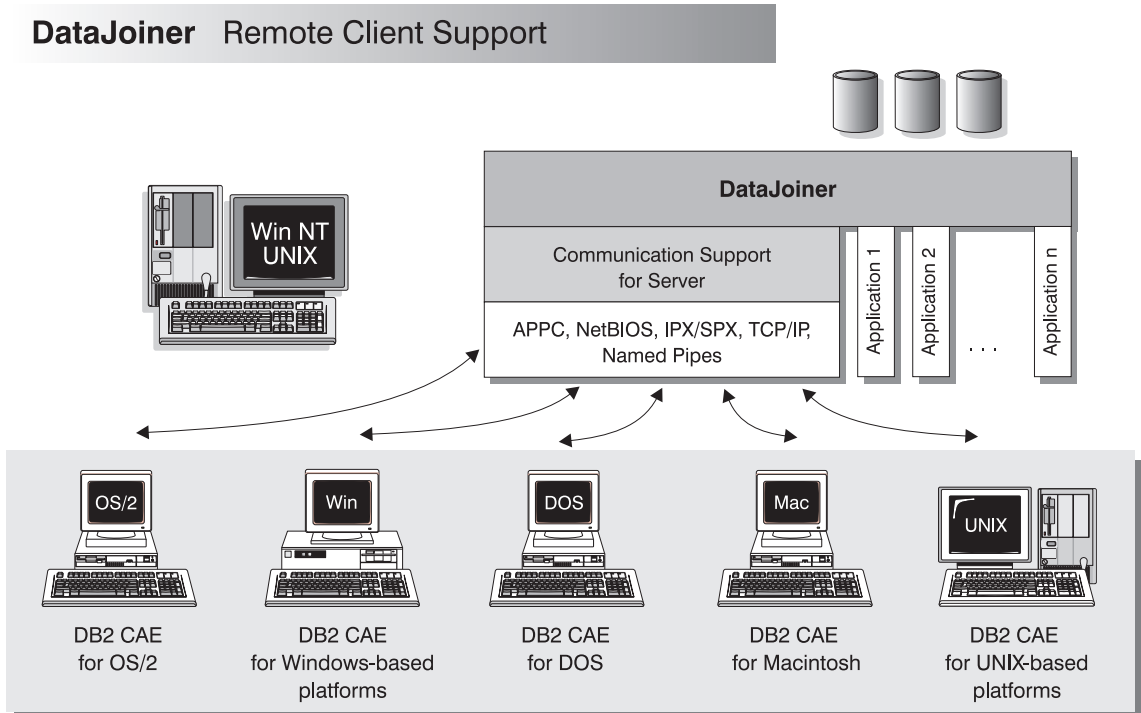
The DB2 Client Application Enabler provides a run-time environment for client workstations that want to access multiple remote databases. Support is available for clients on the following platforms:

- DOS
- Linux
- Macintosh
- OS/2
- UNIX (AIX, HP-UX, Solaris, SINIX)
- Windows NT, Windows 98, Windows 95, and Windows 3.x

In addition to the clients listed above, VM, MVS, OS/400, and any other DRDA clients are supported through the DRDA application server feature. For more information, see the *DB2 Installation and Operation Guide*.

Accessing Database Data From Remote Clients

DataJoiner can be accessed by remote clients that have the DB2 Client Application Enabler installed. Figure 1 shows a DataJoiner server working with local applications and remote clients.



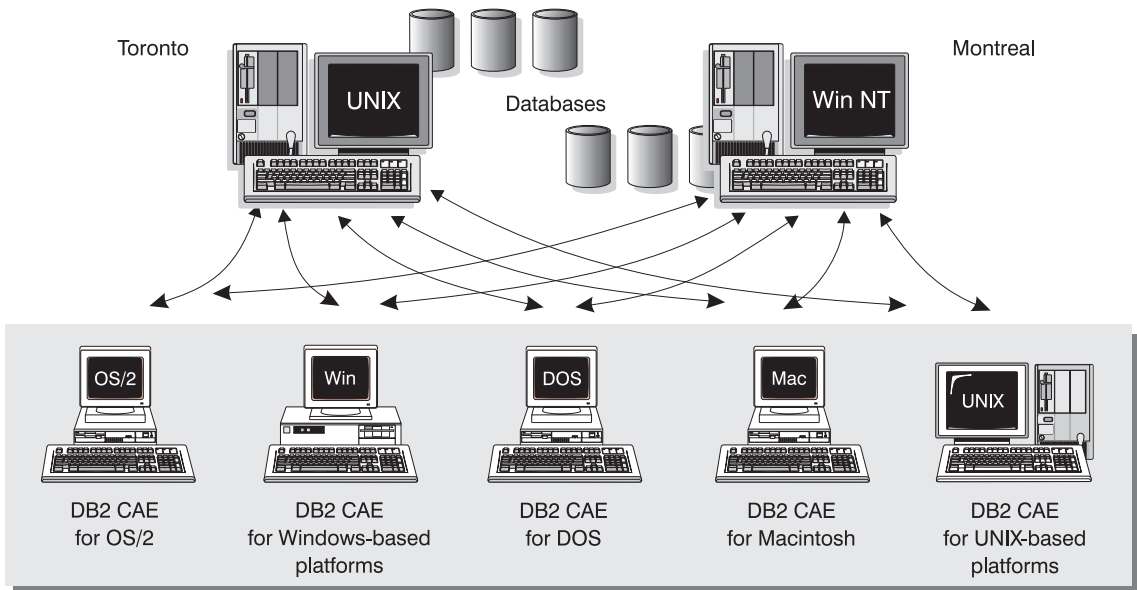
NOTE: CAE is Client Application Enabler. Not all protocols are supported for all the clients.

Figure 1. DataJoiner Server

Accessing Multiple DataJoiner Servers

No additional software is required to enable LAN-to-LAN connections between DataJoiner clients and DataJoiner servers. Suppose you have a DataJoiner server on a Windows NT workstation connected to a LAN located in Montreal and a DataJoiner server on a UNIX workstation connected to a LAN located in Toronto. If there is a connection between the two LANs, clients on either network can access either server. See Figure 2.

Communicating between DataJoiner Servers



NOTE: CAE is Client Application Enabler.

Figure 2. Accessing Multiple DataJoiner Servers

Two-phase commit, or distributed-unit-of-work access, is also supported. Within a single transaction, databases on multiple servers can be accessed and updated with full integrity. See the *DB2 Information and Concepts Guide* for more information about two-phase commits.

Database Administration Tools

You can perform database administration tasks locally or remotely with a collection of tools that helps database administrators (DBAs) manage databases. These tools include:

- The Command Line Processor
- The Database Director
- Visual Explain
- The DB2 Performance Monitor

These tools are described in the following sections.

The Command Line Processor

The Command Line Processor (CLP) is part of DataJoiner base functionality. Use the CLP to enter commands and SQL statements.

Database Director

The Database Director is included in the Administrator's Toolkit. It is an easy-to-use graphical interface that displays database objects (such as databases, tables, and packages) and their relationship to each other. Using the Database Director, you can select one or more database objects and perform the following tasks:

- Configure databases and database manager instances
- Manage the directories necessary for accessing local and remote databases
- Back up and recover databases or table spaces
- Manage table spaces
- Launch Visual Explain (see “Visual Explain” for information)
- Launch the DB2 Performance Monitor (see “DB2 Performance Monitor” on page 7 for information).

Visual Explain

Visual Explain is included in the Administrator's Toolkit. It is a tool for analyzing and tuning SQL statements. This tool helps database administrators and application developers:

- View the access plan chosen by the database manager's optimizer for a given SQL statement.
- View the details of the access plan including statistics in the system catalogs.
- Determine the source of problems in SQL statements.

“Tuning SQL Statements Using Visual Explain” on page 141 provides an example of using Visual Explain to analyze an SQL query. The online help Visual Explain contains additional information.

DB2 Performance Monitor

The DB2 Performance Monitor is included in the Administrator's Toolkit. Use it to help you monitor the performance of your DataJoiner system for tuning purposes. With the DB2 Performance Monitor, you can:

- Determine and analyze performance problems in the database manager or database applications.
- Tune SQL statements for better performance.
- Identify exception conditions based on thresholds that you define.

- Define your own statistics, in addition to the default set provided.

You can choose to monitor snapshots or events. Snapshot monitoring captures point-in-time information at specified intervals. Event monitoring records performance information over the duration of an event such as a connection.

Application Development Using the Software Developer's Kit

The Software Developer's Kit is a collection of tools for database application developers. The Software Developer's Kit includes programming libraries, header files, online documentation, and sample programs to build character-based, multimedia, or object-oriented applications.

Software Developer's Kit tools support several programming languages (including COBOL, C, and C++) for application development, and precompilers for the supported languages are provided.

The Software Developer's Kit can be installed with the DataJoiner server or on a separate machine to provide an application development environment from both server and client platforms. When installed on the server, it provides access to both remote and local databases. When installed on the client, it provides access to remote databases.

There is a platform-specific version of the Software Developer's Kit available for each of the supported client and server environments. The DataJoiner CD-ROM contains Software Developer's Kit functionality for Windows NT. Applications developed with it will run on any client platform where the equivalent DB2 Client Application Enabler component is installed, and can access all DataJoiner and DB2 Servers as well as any other database server that implements the DRDA protocol.

The Software Developer's Kit allows you to develop applications using the following interfaces:

- Embedded SQL
- Call level interface development environment (compatible with Microsoft's ODBC)
- Application programming interfaces (APIs) to access database utilities
- A prototyping environment using the command line processor's interactive SQL

See the *DB2 SDK Building Your Applications* book for your platform for complete information on using the Software Developer's Kit, or for a list of the supported compilers.

Internet Connection Using Net.Data

The fast-growing popularity of the Internet and the World Wide Web has created a demand for Web access to enterprise data. Net.Data is a Web server gateway to IBM's DB2 family. It enables an application developer to build Web applications that access DB2 databases by using HTML forms and dynamic SQL. These applications are then stored on the Web server, and the end user can use any Web browser to access the HTML forms and the reports returning DB2 data.

A DataJoiner application built with Net.Data displays a typical Web page form (an HTML document) to the end user who can select values from a list or type in values to define the query. The user then clicks a push button to submit the query to the Net.Data run-time engine on the Web server. At the Web server, the complete SQL statement is dynamically built with the user inputs and then sent to the database. If user input is not required for the query, a link in the HTML document can trigger a prewritten SQL query and display the resulting report. This type of automated query could be used for repeated access to current DataJoiner data.

A live demonstration of several Net.Data applications is available for frame-enabled browsers on the Web at URL <http://www.ibm.com/software/>. (This address is for the IBM Software Page. From here, select the Data Management link to find the Net.Data information.) The applications for this demo allow customers to query the CelDial Company database for the status of their accounts, to update account information, and to retrieve information about pending orders.

Net.Data can access databases locally, if it is installed with the DataJoiner server or IBM Internet Connection Server. To run Net.Data as a remote client to a DataJoiner server, you need to install Net.Data and IBM Internet Connection Server on a client workstation. See Figure 3 for an example of this scenario.

Accessing DataJoiner Data Using Net.Data

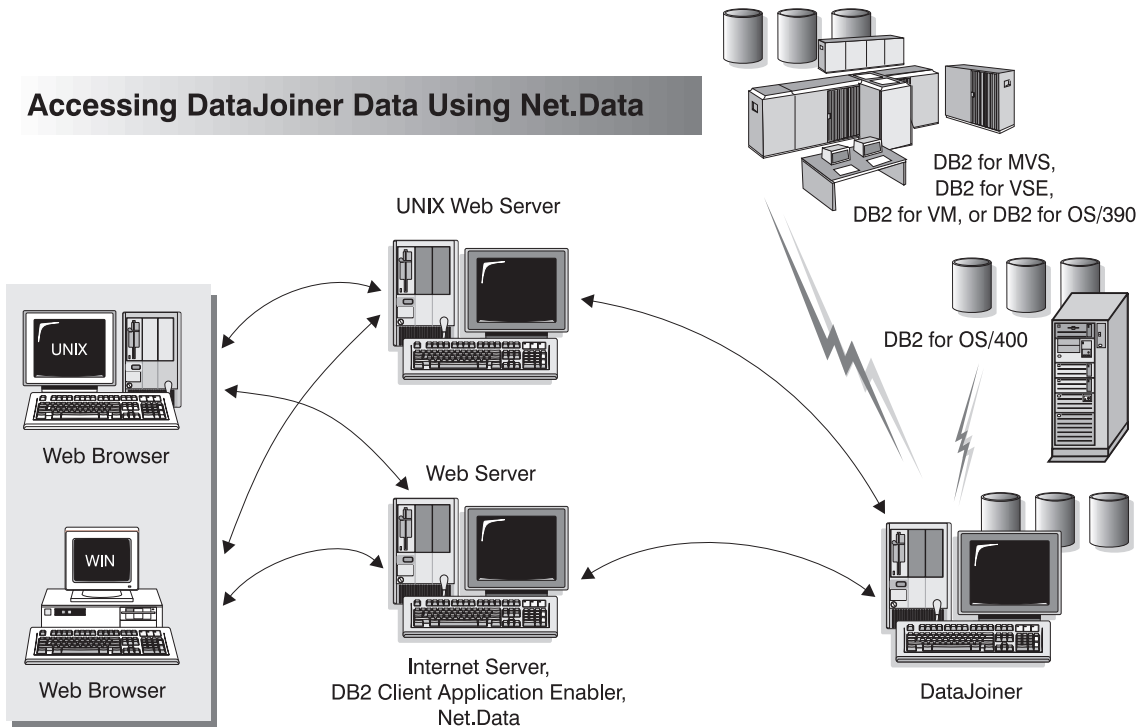


Figure 3. Net.Data

Host Data Access from the Desktop Using DataJoiner DDCS

DataJoiner contains Distributed Database Connection Services (DDCS) functionality, which gives clients on your LAN access to data stored in host systems. A huge amount of organizational data is managed by DB2 on systems such as DB2 for MVS, DB2 for VSE or VM, and DB2 for OS/400. Applications running on any of the supported client platforms can work with this data transparently as if it were managed by a local database server.

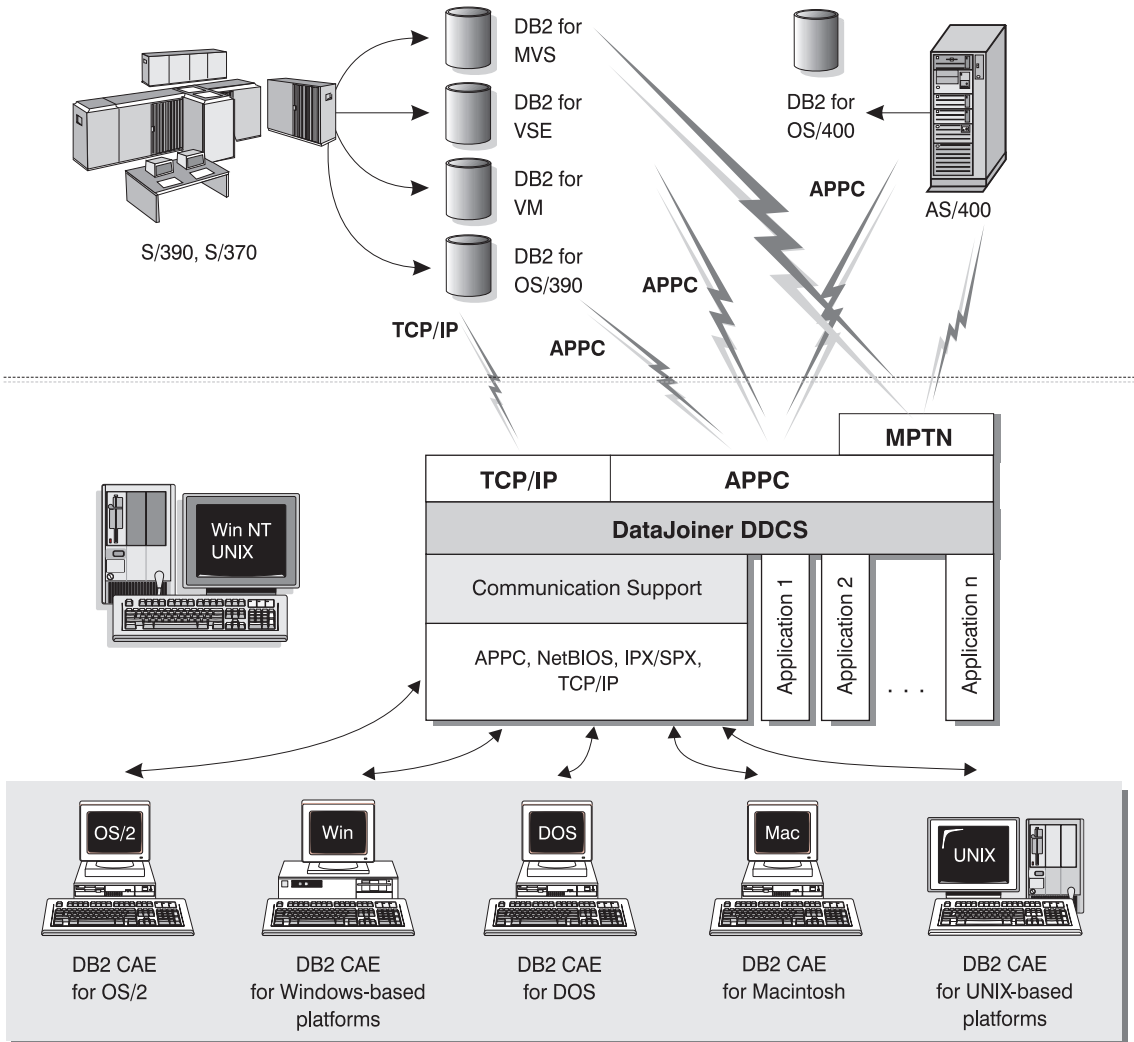
DataJoiner DDCS provides applications with transparent online access to data by implementing a standard architecture for managing distributed data. This standard is known as Distributed Relational Database Architecture (DRDA). For example, you can use DataJoiner DDCS with:

- Spreadsheets, such as Lotus 1-2-3 and Microsoft Excel, to analyze real-time data without the cost and complexity of data extract and import procedures.
- Decision support tools, such as Intersolv Q+E Database Editor, and Crystal Reports, to provide real-time information.
- Personal database products, such as Lotus Approach and Microsoft Access.

- Development tools, such as IBM VisualAge C or C++, Micro Focus COBOL, Microsoft Visual C or Visual C++, to create client/server solutions.
- The Replication Administration Tool and the Apply program to propagate data from or to DB2 systems. DataJoiner's nickname function is used only for access to non-DB2 systems.

DataJoiner DDCS provides the ability for multiple clients to connect to host data. By concentrating definitions and configurations for host access, this configuration can significantly reduce the effort required for implementing and administering access to enterprise data. See Figure 4.

DataJoiner DDCS



NOTE: CAE is Client Application Enabler. Not all protocols are supported for all the clients.

Figure 4. DataJoiner DDCS

DataJoiner DDCS allows DB2 systems on the host to assume the role of an enterprise super server. This solution allows for established data management procedures to be maintained while applications are distributed to workstation platforms to take advantage of graphical user interfaces, distributed processing power, and excellent development tools.

The DDCS functionality included with DataJoiner is equivalent to DDCS 2.3.2 with additional functionality added that allows DataJoiner to access DRDA data sources using TCP/IP.

Operating System Security

DataJoiner for Windows NT uses the Windows NT security services to:

- Determine the identity of the user that is currently logged on.
- Find out, based on a user ID, whether the user is a system administrator. The user can be the connecting user or the binder of the program that is being executed.
- Enumerate, based on a user ID, all the groups to which a user belongs.
- Verify a user ID and password as provided on a CONNECT statement.

To ensure that system objects are appropriately protected, DataJoiner for Windows NT provides user and group security in the following areas:

- Local and domain database access
- Remote database access
- User ID authorization level verification

Local and Domain Database Access

Before you can access the desktop, you must log onto a workstation or domain. The operating system must authenticate your user ID before granting you access to perform database activities.

Local database access means that you are known only locally, that is, to a specific workstation. When you log onto a workstation, the operating system grants you access to that workstation. If you attempt to access a remote database that has server authentication, a user ID and password are required.

Domain database access means that you can access domain databases that have client or server authentication. Logging onto a domain is the same as logging onto a server in a multi-user operating environment. With domain access, you can take advantage of the NT feature of *Trusted domains*. This allows one domain to grant another domain access to its resources and users from one domain to access another domain without being required to logon to each domain separately.

Remote Database Access

User access to remote databases is provided as follows:

- If the server instance has client authentication, a user ID and password are not required.

- If the server instance has server authentication, a user ID and password are required even if the user has already logged on.
- If the server instance has Trusted Client Authentication, a user ID and password are required only if the remote NT server considers the client an *Untrusted* client.

Notes

- DataJoiner for Windows NT regards remote instances with client authentication as Trusted clients. Trusted clients are clients that have security systems. The following clients are considered Trusted clients:
 - AIX
 - OS/400
 - HP-UX
 - MVS
 - OS/2
 - Solaris
 - VM
 - Windows NT
- Untrusted clients, such as DOS, Windows, and Windows 95 clients, require a user ID and password, even when the authentication of the DataJoiner instance is "Client."

User ID Authorization Level Verification

The DataJoiner for Windows NT security model uses the Windows NT Security Access Management (SAM) database to obtain user and group information, user ID authorization, and user ID-password validation. However, DataJoiner for Windows NT has the following restrictions:

- User IDs, group IDs, and passwords are limited to a maximum of eight characters.
- User IDs and group IDs must be valid SQL identifiers.

Heterogeneous Data Replication

With DataJoiner and the Apply program, you can propagate data between DB2 systems, other relational databases, and non-IBM relational databases. DataJoiner's Apply program is equivalent to the Apply program available with DB2 Universal Database Version 5. DataJoiner's replication administration tool is an administrative interface to help you manage replication operations from a single point of control. The replication administration tool runs on a Windows 95 or Windows NT client workstation.

For the replication administration tool and the Apply program to communicate with DB2 data sources, you need to configure DataJoiner DDCS; for the replication administration tool and the Apply program to communicate

with non-DB2 data sources, you need to configure server and user mappings. See “Chapter 7. Configuring Access to Data Sources” on page 51 for more information.

Differences in the DataJoiner for Windows NT Environment

In the Windows NT environment, the following features are different:

The Command Line Processor

You cannot use the DataJoiner for Windows NT command line processor from any command window. If you attempt to use the command line processor without preparing a command line environment, you will receive the following error message:

```
DB21060E Command line environment not initialized.
```

To initialize the environment and create a DB2 command window, do one of the following:

- Invoke a DB2 command window from the icons provided in the DataJoiner for Windows NT program group.
- Enter the command **db2cmd**.

The Performance Monitor

DataJoiner for Windows NT interfaces with the Windows NT Performance Monitor to capture performance data on all aspects of DataJoiner. The Performance Monitor displays performance information in a variety of forms. For example, you can retrieve information on any of the following objects:

- DataJoiner for Windows NT database manager for general server information. The DataJoiner instance that you are monitoring is shown as the object instance.
- DataJoiner for Windows NT databases for information on a specific database. An instance is displayed for each currently active database.
- DataJoiner for Windows NT applications for information on an active DataJoiner application. An instance is displayed for each currently active DataJoiner application.

Windows NT Event Log

The DB2 First Failure Support is implemented using the Windows NT Event Log system. It provides failure analysis support functions, such as error detection and message logging during the operation of certain DataJoiner components.

The Windows NT Security Subsystem

DataJoiner for Windows NT takes advantage of the Windows NT

security services to provide you with flexibility in controlling your security environment. See “Operating System Security” on page 13 for more information about DataJoiner for Windows NT security.

RAW I/O Support

DataJoiner’s support of RAW I/O allows you to attach a raw device to an NT system. To attach a device, use the following device naming convention:

```
\\.\PhysicalDrivex
```

where x is the device ID for the first drive.

For a complete discussion of the use of RAW/IO with DataJoiner for Windows NT, contact IBM through **CompuServe**.

Code Page Support

The Japanese version of Windows NT uses the ANSI code page 932. However, the IBM National Language Centers NT system have determined that the Japanese 943 code page is equivalent to the Windows NT 932 code page.

To ensure that codepoints are not advertently lost when communicating with other DataJoiner clients or servers that use the Windows NT 943 code page, DataJoiner maps the 932 code page to 943, thereby ensuring that 943 becomes the default on the Japanese version of Windows NT.

DB2 host databases that are accessible through IBM’s DDCS gateways do not currently support the 943 code page. To communicate with host databases using the Japanese version of Windows NT, the environment variable DB2CODEPAGE must be used to override the default value of 943. For example, you can set the DB2CODEPAGE environment variable to the 932 code page that the host supports.

Chapter 2. What's New in DataJoiner Version 2?

DataJoiner Version 2 offers new features and enhancements, including:

DB2 Version 2 functionality

DataJoiner is built on the DB2 Version 2 code base, which means that DataJoiner provides all the major functional enhancements provided by DB2, including:

- Extended SQL capabilities
- An enhanced SQL optimizer
- Improved database performance
- Systems management support
- Robust integrity and data protection
- Object relational capabilities
- National language support (NLS)
- Support for the Java Development Kit (JDK) 1.1 for the Java Database Connectivity (JDBC) API
- DDL transparency

For detailed information about many of these features, see the *DB2 Administration Guide*.

DataJoiner for Solaris systems

DataJoiner now runs on the Sun Solaris operating system. DataJoiner for Solaris systems supports the same SQL as DataJoiner for AIX systems.

DataJoiner for Windows NT systems

DataJoiner provides industrial strength heterogeneous database management on Windows NT systems. DataJoiner for Windows NT systems supports the same SQL and features as DataJoiner for AIX systems.

Support for Oracle 8, RDB, SQL Anywhere, and Teradata

With Version 2, DataJoiner continues to increase the number of natively supported data sources. The most recent additions are:

- Oracle 8
- Oracle RDB Version 6 and above (on DataJoiner for Windows NT systems only)
- Sybase SQL Anywhere Version 5.0 (on DataJoiner for Windows NT systems only)
- Teradata (on DataJoiner for AIX and DataJoiner for Windows NT systems only)

Spatial Extender

DataJoiner now supports geographic information system (GIS) data (also known as spatial or geographic data). New data types, spatially-enabled columns, and spatial join capability allow you to take advantage of geographic data in your applications. Included are powerful two-dimensional functions that allow you to create specific relationships among the geographic objects you define. Included with the spatial extender are the following components:

- A set of spatial data types
- A set of spatial operations and predicates
- A set of spatial index data types
- An administration tool suite for the spatial extender
- Sample programs

You can also take advantage of existing geographic data stores using the load and transform capability of the Spatial Extender.

Expanded DataJoiner SQL support

This version of DataJoiner contains many new and modified SQL statements. New DDL statements provide greater flexibility and safety in defining your DataJoiner environment—users can create, alter, and drop mappings for data sources, users, user-defined and built-in functions, and data types. Additionally, new SQL DML statements provide enhanced functions for local and distributed queries; an example is the CASE expression, which is useful for selecting an expression based on the evaluation of one or more conditions.

DataJoiner SQL for creating, altering, and deleting data source tables

Version 2 includes a new DataJoiner SQL statement for creating tables in different types of data sources. If the native SQL for creating tables in these data sources includes a unique option—for example, the option in DB2 for OS/390 for specifying what database you want a table to reside in—you can code this option in the new DataJoiner statement. If you create a data source table with this new statement, you can also alter and delete it with DataJoiner SQL.

Heterogeneous data replication

DataJoiner now provides replication administration as an integrated component. You can define, automate, and manage replication data from a single control point across your enterprise. The replication administration tool provides administrative support for the replication environment, with objects and actions that define and manage source and target table definitions. DataJoiner's Apply component performs the actual replication, tailoring and enhancing data as you specify, and serving as the interface point to and from your various data sources. DataJoiner also supplies an executable, IBM DB2 DataPropagator for

Microsoft Jet, that allows you to replicate server data for browsing and updating in LAN, occasionally connected, and mobile environments.

Distributed heterogeneous update support

DataJoiner for AIX and DataJoiner for Windows NT systems now allow you to update multiple heterogeneous data sources within a distributed unit of work while maintaining transaction atomicity. This task is accomplished through adherence to the two-phase commit model. Supported data sources include most versions of the DB2 Family and, with the appropriate XA libraries, various other data sources as well.

New graphical installation, configuration, and administration tools

A variety of new tools is available to help you accomplish administrative chores. Wizards walk you through data source configuration. And the Administrator's Toolkit provides a collection of tools designed to assist you with the day-to-day operation of DataJoiner. It includes the following components:

The Database Director

Allows you to perform configuration, backup and recovery, directory management, and media management tasks.

Visual Explain

A tool for graphically viewing and navigating complex SQL access plans.

The DB2 Performance Monitor

Monitors the performance of your DB2 system for tuning purposes.

The Administrator's Toolkit is available with DataJoiner for AIX and DataJoiner for Windows NT systems.

Stored procedures

DataJoiner now supports stored procedures at remote data sources as well as the local DataJoiner database. Use stored procedures to speed application performance. For example, applications that process huge amounts of data at a server but return smaller result sets should run faster as stored procedures. Another benefit is that stored procedures usually reduce network traffic between clients and databases.

DataJoiner stored procedures can augment standard data security. For example, in a 3-tier environment, data can be retrieved from a remote server and then processed at the DataJoiner server; only a subset of data needs to be available to the client.

System catalog information available in views

DataJoiner provides views from which you can access system catalog

information about each DataJoiner database. Some of these views contain data—for example, data about tables, indexes, and servers—that was accessible only from tables in previous versions of DataJoiner. Other views contain data—for example, data about stored procedures, server options, and server functions—that is now available in Version 2.

Performance enhancements

In addition to general engine performance improvements, this latest version offers new query rewrite capabilities, improved pushdown performance, and remote query caching.

Part 2. Planning for and Installing DataJoiner

Chapter 3. Planning for Installation



If you already know that your system meets DataJoiner's hardware and software requirements, and you want to start installing DataJoiner, go directly to "Installing from CD-ROM" on page 33.

DataJoiner provides many components and features that you might want to use in your environment. This chapter provides the following product information and planning information for DataJoiner:

- "Hardware Requirements"
- "Software Requirements" on page 25
- "Client Access to DataJoiner" on page 27
- "Supported Compilers for DataJoiner" on page 28
- "Replication Considerations" on page 28

Hardware Requirements

This section shows the minimum amount of disk space that is required and the minimum memory usage of each DataJoiner component. This information is helpful in planning for more sophisticated systems with a large number of concurrent clients or databases.

The amount of disk space and the amount of memory that is listed here are averages. The actual amounts that are required depend on the functions you use.

To install DataJoiner, you need:

- An Intel® type microprocessor-based system. DataJoiner for Windows NT does not run on RISC-based systems.
- A CD-ROM reader.
- At least 100 MB of hard disk space (for DataJoiner only).
- A minimum of 16 MB of working set memory.

For replication requirements, see "Chapter 25. Replication Requirements" on page 197.

Requirements for the operating system, application development tools, and communications products are not included in this chapter. Consult the individual product documentation for these values.

Memory and Fixed Disk Estimating Table

Use Table 1 to estimate the amount of memory and disk space you need to install DataJoiner on your Windows NT server systems.

To estimate the memory requirements and disk requirements for a particular configuration, add the recommended memory size for each installed function and application that is to be run concurrently. Include an allowance for your application data requirements and buffer requirements.

Working set memory includes the parts of a program's executable code and data areas that are being used intensively. Working set memory figures assume that default values are used.

The memory requirement is less if you do not run all installed functions and applications concurrently. (System memory requirements are based only on applications and operating system functions that run concurrently.)

Having less memory than the calculated recommended memory size can increase the time that is required to switch from one function or application to another, reduce keystroke responsiveness, reduce throughput, and increase the size of the virtual memory paging space that is required.

Table 1. Estimating Disk and Memory Requirements

	Recommended Minimum Working Set Memory (MB)^a	Recommended Minimum Disk (MB)
DataJoiner Server		
DataJoiner (base) ^b	16	100
For first local application	2.2	5.4
For each additional remote client	0.31	0.22
For each additional concurrent database ^c	2	25
For each additional concurrent instance	7	10
Administrator's Toolkit	N/A	24
East Asian Conversion Support	N/A	4.5
SDK Sample Programs	N/A	2
Online documentation in HTML Format (English)	N/A	16

Table 1. Estimating Disk and Memory Requirements (continued)

Table Notes:

- a** Additional memory for some database configuration parameters, including the database buffer pool or the sort heap, might be required depending on your workload.
- b** Assumes one remote process that is using the local database, on one instance. DataJoiner base includes communications support and the Apply program for replication.
- c** These recommended values do not reflect the resources needed to create a database; they reflect resources needed to connect to a database only.
- The extra disk space for each additional database is for table definitions and internal structures for each database and does not include user data; therefore, the amount of extra disk space required is subject to many variables. Actual requirements may differ.
- The default configuration allocates three log files, each 10 KB in size. These logs can be relocated to another file system, and can be made larger or smaller as required. See the *DataJoiner Administration Supplement* for more information.
-

Software Requirements

This section lists the software that is required to run DataJoiner for Windows NT.

Requirements for the DataJoiner Server

This section lists the required operating system and communications software that is supported for DataJoiner.

Operating Systems

- Windows NT Server Version 4.0
- Windows NT Workstation Version 4.0

Communications

APPC, NetBIOS, TCP/IP, or IPX/SPX

Notes:

- Windows NT supplies communication support for TCP/IP, NetBIOS, IPX/SPX, and Named Pipes connections.
- For APPC connectivity, you need one of the following communications products:
 - IBM Communications Server for Windows NT Version 5.0 or later

- IBM Personal Communications for Windows NT Version 4.1 or later
- Microsoft SNA Server Version 2.11 or later
- To install products by using Systems Management Server (SMS), you must already have the SMS product installed and configured.

To determine requirements for DataJoiner's replication components, see "Chapter 25. Replication Requirements" on page 197.

Requirements for DataJoiner Data Sources

Table 2 lists the data sources that DataJoiner supports, as well as any required client software needed to access these data sources.

Table 2. DataJoiner Data Sources

Data Source	Client Software Needed on the DataJoiner System
IBM DataJoiner	included in DataJoiner
IBM DB2 Universal Database V5.0	included in DataJoiner
IBM DB2 for CS V2	included in DataJoiner
<ul style="list-style-type: none"> • IBM DB2 for MVS V2R3 with APAR PN43153, UN75958, UN54600, UN56735 • IBM DB2 for MVS V3R1 with APAR PN70612, UN42626, UN54601, UN56736, UN73393 • IBM DB2 for MVS V4R1 with APAR PN70612 • IBM DB2 for OS/390 	included in DataJoiner
<ul style="list-style-type: none"> • IBM DB2 for OS/400 V2R2 with PTF SF13747, SF13748 • IBM DB2 for OS/400 V3R1 or later 	included in DataJoiner
IBM SQL/DS V3R3 with APAR PN43497 or later	included in DataJoiner
<ul style="list-style-type: none"> • IBM SQL/400 V2R2 with PTFs SF13747 and SF13748; or in a DBCS environment, IBM SQL/400 V2R2 with SF31979 • IBM SQL/400 V3R1 with PTF SF23247 or later; or in a DBCS environment, IBM SQL/400 V3R1 with SF32970 	included in DataJoiner
IBM DB2/PE	included in DataJoiner
IBM DB2 for Solaris V1 and V1.2	included in DataJoiner
IBM DB2 for HP-UX V1.2	included in DataJoiner
Informix OnLine V7.1 or later (non-GLS, -XPS, and -SE versions only)	Informix ESQL/RT or equivalent libraries

Table 2. DataJoiner Data Sources (continued)

Data Source	Client Software Needed on the DataJoiner System
Microsoft SQL Server	MS SQL Server 32-bit ODBC driver
ODBC, X/Open CLI-compliant data sources through the generic access API	MS SQL Server 32-bit ODBC driver
Oracle V7.0.13 or later	Oracle SQL*Net Version 7.3 or Oracle Net8
Oracle RDB	ODBC driver
Sybase V4R6 or later	Sybase Open Client ctlib V11
Sybase SQL Anywhere	ODBC driver
Teradata V2	Teradata WinCLI for Windows NT systems
Other nonrelational data sources	Cross Access Corporation's CrossAccess Data Delivery System

Client Access to DataJoiner

Remote clients must access DataJoiner through one of the DB2 Client Application Enablers (CAEs) and Software Developer's Kits (SDKs) V2.1.1 or later. Because remote clients can run in a variety of operating environments, including OS/2, several UNIX-based environments, DOS, Windows 95, and Windows NT, a platform-specific version of the DB2 CAE and SDK is available for each of these environments.

A DB2 CAE component is built into each of the DB2 for CS products. Also, when you purchase DataJoiner or the server version of DB2, you receive a Client Pack that includes each of the available DB2 CAEs; however, you can obtain the most current versions by downloading them from the DataJoiner Web page (<http://www.ibm.com/software/data/datajoiner>). You can copy the CAEs to the appropriate workstations for each remote client that will connect to the server.

A workstation that has a DB2 CAE and SDK installed can access any DataJoiner or DB2 server by using any of the supported communication protocols, including TCP/IP, IPX/SPX, NetBIOS, and APPC. This support protects your investment in client workstations and allows you to select a server machine that is most appropriate for your database environment.

To determine which protocols can be used with your clients, see *Installing and Using Clients* for your particular platform.

Installation restriction:

Do not attempt to install the DB2 CAE on the same system where DataJoiner for Windows NT is installed.

Supported Compilers for DataJoiner

DataJoiner provides support for C, C++, and COBOL compiled languages. Writing an application in a compiled language gives the programmer a high level of flexibility. For a compiled language, an appropriate precompiler must be available to process the SQL statements. DataJoiner for Windows NT provides C and C++ precompilers. The Micro Focus** COBOL compiler provides a COBOL precompiler for use with DataJoiner.

The application development environment for remote clients is provided by the DB2 SDK V1 or higher. Please check your SDK documentation for an up-to-date list of compilers that are supported by the SDKs.

The specific compilers that are supported by DataJoiner for Windows NT include the following:

- Microsoft Visual C++ Version 1.5
- Borland C++ Version 4
- Micro Focus COBOL Version 3.1.49 or later

The DB2 SDK products provide an API that application development tool providers can use to offer a DB2 precompiler within their products. For example, IBM PL/I for OS/2 Professional Edition Version 1.2 and IBM PL/I for AIX Version 1.1 include precompilers for DB2 that exploit this interface.

Replication Considerations

If you are planning on using DataJoiner to replicate data, here are some things you will need to know:

- The replication administration tool runs on a Windows 95 or NT workstation and accesses DataJoiner via the DB2 CAE for either of these Windows platforms. This means that you will configure the DataJoiner for Windows NT instance for remote or local client access, based on whether or not the replication administration tool is installed on the same workstation as DataJoiner.
- Access to different types of data sources is achieved as follows:
 - Both the replication administration tool and the Apply program must use DataJoiner's data access modules to access non-IBM data sources. See the appropriate section in "Chapter 7. Configuring Access to Data Sources" on page 51 for details.
 - To access DB2 for CS or DB2 Universal Database data sources, the Apply program can use DataJoiner's DB2RA protocol, while the replication administration tool can access these data sources directly through the CAE. See "Chapter 9. Accessing DB2RA Data Sources" on page 67 for more information.

- For replication that involves access to IBM IMS/ESA non-relational databases, you need DataPropagator NonRelational (DPROPNR).
- To access DB2/MVS data sources, Apply should go through DataJoiner DDCS, while the replication administration tool can use DataJoiner DDCS, DDCS on another workstation, or DB2 Connect on another workstation.

Consider network performance when deciding how to access data sources. For example, using DDCS for Windows NT on the same workstation as the replication administration tool typically provides faster access to DB2/MVS than going to another workstation to access DataJoiner DDCS and subsequently DB2/MVS. This is dependent on your site's network.

- When replicating from one or more non-IBM relational databases, one DataJoiner database is required for each non-IBM database. Plan accordingly for the file space and memory impact. When replicating to one or more non-IBM relational databases, all can be accessed through a single DataJoiner database, thus saving resources.
- When replicating from non-IBM databases to DB2/MVS databases, Apply/MVS typically is used to replicate data. If Apply/MVS is to be used rather than the Apply component within DataJoiner, configure DataJoiner as a DRDA application server.
- When replicating from Oracle databases, the DataJoiner database you create to access Oracle must be created with the COLLATE USING IDENTITY clause.
- For existing replication installations that used DataPropagator Relational (DPROPR) Version 1, there is no migration program to IBM Replication (the replication tool used in DB2 UDB Version 5).¹ You must use DataJoiner's replication administration tool to set up your replication configuration.
- The DPROPR V1 Capture program cannot be used with the DataJoiner Apply program; you must use the DB2 DataPropagator V5 Capture program.

License Information

When you purchase DataJoiner for Windows NT, you receive a permanent license to use DataJoiner for Windows NT.

1. Sometimes referred to as V5 Apply and V5 Capture programs.

Chapter 4. Installing DataJoiner

This chapter guides you through the DataJoiner installation process on the Windows NT operating system and includes the following sections:

- “Installation Notes”
- “Creating a DataJoiner Administrator User Name” on page 32
- “Installing from CD-ROM” on page 33
- “Verifying the Installation” on page 34

To learn how to solve problems that arise during installation, see “Handling Installation Errors” on page 279.

The process of installing the client products and components is separate from installing the servers. For information about installing clients, see *Installing and Using Clients* for your platform.

Installation Notes

Before installing DataJoiner, note that:

- The README.CD file on the DataJoiner CD-ROM contains the most current installation notes. This file is in the \i386\datajoinernt\en directory. Read this file before attempting to install DataJoiner.
- The process for migrating from a previous release of DataJoiner is different than the process of installing DataJoiner for the first time. If you are migrating from a previous release, please see “Chapter 5. Migrating from Previous Versions” on page 37 for more information.
- The Setup program attempts to replace certain files in the Windows system directory if the current versions of these files are lower than the versions that are required by DataJoiner. However, if a file is currently in use by another application or service, the Setup program will be unable to replace it. The Setup program logs the error message returned by the operating system during the failed copy operation and notifies you that an error occurred.

To remedy this situation:

1. Shut down any other running applications.
 2. Stop any DB2 and DataJoiner services that are running.
 3. Run the Setup program again.
- The system keeps a log (DB2.LOG) that tracks installation activities and uninstall activities and records information on any errors encountered

during product installation or uninstall. The log is stored in the DB2LOG directory. If this directory exists, the Setup program uses it. If not, the Setup program creates the directory on the drive that contains the Windows directory. If space constraints or other errors prevent the log from being opened or written to, the installation or uninstallation continues.

See “Handling Installation Errors” on page 279 for more information.

- You cannot have two versions of a DB2 product installed on the same Windows NT workstation. Therefore, do not install DataJoiner on an NT workstation where a DB2 product already exists.
- DataJoiner, its documentation, and its Setup program are available in the following languages:

EN	English
DE	German
ES	Spanish
FR	French
IT	Italian
JA	Japanese
KO	Korean
PT	Brazilian Portuguese
TC	Traditional Chinese
ZH	Simplified Chinese

The two-letter language codes are used as directory names on the DataJoiner CD-ROM.

Creating a DataJoiner Administrator User Name

Before you install DataJoiner for Windows NT, you need to create a special Windows NT user name that meets DB2’s naming requirements. You must use this user name to install DataJoiner and perform DB2 administration tasks.

The user name that you specify must follow these guidelines:

- It must be a member of the Administration group on the local machine.
- It must be acceptable to DB2. A valid DB2 user name follows these rules:
 - It can contain up to 8 characters.
 - It can include letters, numbers, @, #, or \$.
 - It cannot begin with IBM, SYS, SQL, or a number.

- It cannot be a DB2 reserved word (USERS, ADMINS, GUESTS, PUBLIC, or LOCAL) or an SQL reserved word, which are listed in the *DataJoiner Application Programming and SQL Reference Supplement*.
- It cannot end with \$.
- It cannot include accented characters.

Restriction:

Do not attempt to install or administer from Windows NT's default "Administrator" user name. Because that user name has more than eight characters, DB2 does not accept the request.

To create a user name:

1. Click the **Start** button and select **Programs->Administrative Tools (Common)->User Manager**. The User Manager window opens.
2. Select **User->New User** from the menu bar and create a new user that belongs to the Administrator group on the local machine.
3. You are now ready to install DataJoiner. Log off the system and log on with the user name that you just created.

Installing from CD-ROM

Before installing from CD-ROM, refer to "Installation Notes" on page 31.

1. Log on to Windows NT with a valid user name that has administrator authority (see "Creating a DataJoiner Administrator User Name" on page 32).
2. Insert the DataJoiner CD-ROM into the appropriate drive. The Setup window opens, and the first DataJoiner Installation screen is displayed. Alternately, you can initiate the Setup program by using either of the following methods:

- Using Windows NT Explorer:
 - a. Open Windows NT Explorer.
 - b. Double-click the icon that represents your CD-ROM drive.
 - c. Double-click the **i386** folder.
 - d. Double-click **Setup**.

- From a command line, type the following command:

```
cdrom_drive:\i386\SETUP.exe
```

where *cdrom_drive* is the letter that designates your CD-ROM drive.

3. Select the DataJoiner products you wish to install, and respond to the Setup program's prompts. Selectable products include:
 - DataJoiner for Windows NT

- Replication Administration
- DataJoiner Jet Replication
- Classic Connect data mapper for IMS and VSAM
- DB2 Spatial Extender

In a multi-product installation, DataJoiner is installed last because it is the only product that requires you to reboot your system.

Replication Restriction:

You can install the replication administration tool without installing DataJoiner. The Apply component, however, is part of DataJoiner basic functionality and is not separately installable.

If you want to install only particular DataJoiner product components, choose a custom installation. Selectable components include:

- Sample Applications
 - Documentation
 - East Asian Conversion Support
4. Continue with the installation by following the TaskGuide’s prompts. As the installation proceeds, note *all* informational or error messages that are displayed. See “Handling Installation Errors” on page 279 for the action to take if an error occurs.
 5. At the end of the installation, reboot your workstation.

Verifying the Installation

After installing DataJoiner, you might want to verify that the installation completed successfully. The following sections explain how to do so.

Verifying Instances

Verify that a default DataJoiner instance and the DB2 Security Server have been created and registered as services:

1. Double-click **My Computer**.
2. Double-click **Control Panel**.
3. Double-click **Services**, and verify that the following two items are present in the list:
 - **DB2-DB2** (default DataJoiner instance)
 - **DB2 Security Server**



From here, you can specify whether or not DataJoiner is started automatically by double-clicking these two services and changing the **Startup Type** to **Automatic**.

Checking the Programs Menu

Verify that **DataJoiner for Windows NT** appears in the **Programs** menu:

1. Click the **Start** button.
2. Select **Programs**.
3. Verify that **DataJoiner for Windows NT** is present in the list.

Creating the Sample Database and Selecting Sample Data

A sample database, called SAMPLE, is provided with DataJoiner. The sample database contains tables, views, and fictitious data. Many examples that are provided in this book and in the product documentation use these sample tables to show you how to use the functions that are described. For a detailed description of the sample database, see the *DB2 Administration Guide*.

To create the sample database:

1. Initialize the DB2 command line environment by typing **db2cmd** at a command prompt.
2. Enter the **db2sampl** command.

Optionally, you can create the database on a different drive. For example, type the following command to create SAMPLE on the F drive:

```
DB2SAMPL F:
```

Notes about creating the sample database:

- This command might take a few minutes to process. There is no completion message; when the command prompt returns, the process is complete.
 - The sample database requires 8 MB of disk space.
 - The database is automatically cataloged with an alias of SAMPLE when it is created.
 - When you do not need to access SAMPLE and want to remove it, type

```
DB2 DROP DATABASE SAMPLE
```
 - To create your own database, use the **CREATE DATABASE** command; see “Creating a DataJoiner Database” on page 43 for more information.
3. Select some sample data:
 - a. At a command line, enter:

```
DB2 connect to sample
```
 - b. Issue the following SQL statement:

```
DB2 select * from staff where ID = 220
```

You should receive the following result set:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
220	Smith	51	Sales	7	17654.50	992.80

If you can complete all the steps in this section, you have successfully installed DataJoiner.



- After successfully installing DataJoiner, Spatial Extender users should refer to *IBM DB2 Spatial Extender Administration Guide and Reference* for further configuration instructions.
 - If you want to learn more about basic DataJoiner tasks; see “Chapter 6. Getting Started with DataJoiner” on page 41.
 - You also can begin to configure data sources now; see “Chapter 7. Configuring Access to Data Sources” on page 51 for instructions.
-

Chapter 5. Migrating from Previous Versions

Migrating from Version 2.1 of DataJoiner for Windows NT to Version 2.1.1 involves the following procedures:

- “Migrating Instances”
- “Migrating Databases”

CAUTION!

All migrations inherently involve a certain amount of risk, and DataJoiner is no exception. Therefore, proceed with caution. Read this entire chapter before starting the migration process, and make appropriate backups.

Migrating Instances

Version 2.1 instances are dynamically migrated to Version 2.1.1 during installation. When you attempt to install DataJoiner V2.1.1 on a machine where DataJoiner V2.1 is installed, the following sequence of events occurs:

1. All local databases cataloged in the instance are checked for conditions that would prevent successful migration.
2. If any of these conditions are detected, a log file is generated that lists those conditions. This file is named DB2.LOG and is contained in the DB2LOG directory.

Correct all reported errors by using the information in “Correcting Migration Inhibitors” on page 38.

If no errors are found, the instance migration is initiated.

Migrating Databases

Use the following information to migrate your databases. The database manager must be started before migration can begin.

Restrictions:

You cannot migrate a database that is in one of the following states:

- Backup pending
- Roll-forward pending
- Database inconsistent

In addition, you cannot migrate a database that contains any database objects that have a qualifier (schema name) of SYSCAT or SYSSTAT. These two schema names are reserved for use by the database manager.

See “Correcting Migration Inhibitors” for further instructions.

1. Backup your database.

CAUTION!

Database migration is not a recoverable process. If you backup your database before the schema names are changed, you will not be able to restore the database from backup by using DataJoiner Version 2.1.1. You will have to use the version of the database manager from which you are migrating your databases.

If you do not have a backup of your database from before you attempted migration, and the migration failed, you have no way of restoring your database by using DataJoiner Version 2.1.1 or your previous version of the database manager.

2. Migrate the database by using one of the following:
 - The **MIGRATE DATABASE** command-line processor command
 - The **RESTORE DATABASE** command, when restoring a full backup of the database

“Selective Migration” on page 39 describes what to do if you do not want to migrate all databases in a given instance.

3. Rebind all packages. All existing packages are marked as invalid during catalog migration. You can use the **DB2RBIND** utility to revalidate all packages, or allow package revalidation to occur implicitly when a package is first used. The **REBIND PACKAGE** or **BIND** commands selectively bind a particular package.
4. Tune your database and database manager configuration parameters to take advantage of Version 2.1.1 enhancements.

Correcting Migration Inhibitors

The following table shows common inhibitors to successful database migration and prescribes appropriate solutions.

Table 3. Common Migration Inhibitors and Recommended Solutions

State of Database	Corrective Action
Backup pending	Backup the database.
Roll-forward pending	Recover the database as required. Perform or resume a ROLLFORWARD DATABASE .

Table 3. Common Migration Inhibitors and Recommended Solutions (continued)

State of Database	Corrective Action
Database inconsistent	Restart the database to return it to a consistent state.

See the (Version 2) *DB2 Administration Guide* for more information about the actions that are required to correct these conditions.

Requirement:

If databases contain one or more objects that use SYSCAT or SYSSTAT as schema names, then these objects must be dropped and recreated using a different schema name. Therefore, we recommend that you avoid using schema names that begin with SYS altogether.

All local databases now have the same authentication type as the instance where they reside. The authentication type in the database directory is ignored by DataJoiner Version 2.1.1 servers. If a warning is logged due to a *conflicting* authentication type, you have the following options to retain the same authentication type in your database:

- Change the authentication type of the instance.
- Move the database to another instance that has the desired authentication type.

Before changing the authentication type of the instance, you should make sure that the new authentication type will be appropriate for all databases that reside there. Be sure to consider the security implications of the different authentication types.

Suggestion:

If there are databases that you do not want to migrate, you can uncatalog them (along with all aliases).

Selective Migration

To migrate only certain databases under a given instance, first execute the **db2ckmig** command to verify that the selected databases can be migrated.

1. Login as sysadm.
2. Ensure that all local databases that you wish to migrate are cataloged before running the **db2ckmig** command.
3. End all applications that are currently using database manager.
4. Stop the database manager and the command line processor.
5. Issue the **db2stop** and **db2 terminate** commands.
6. Execute the **db2ckmig** command as follows:

►—db2ckmig—database—/l filename—◄

where:

database

Specifies an alias name of a database to be scanned

/l Specifies a file to be used to keep a log of conditions that would prevent the successful migration of the database

An error is logged if the database is in one of the following states:

- Backup pending
- Roll-forward pending
- Database inconsistent

An error is also logged for each database object that uses either SYSCAT or SYSSTAT as the qualified object name.

Restriction:

SYSCAT and SYSSTAT are reserved schema names in Version 2.1.1.

The **db2ckmig** parameters may be preceded by either a "-" or "/".

Correct all the errors that are reported (see “Correcting Migration Inhibitors” on page 38 for suggested corrective actions).

Back up the databases and restore them under a newly created Version 2.1.1 instance.

Chapter 6. Getting Started with DataJoiner

After you successfully install DataJoiner, you can begin learning about the everyday tasks that are associated with using DataJoiner. This section describes how to use DataJoiner in a variety of ways. The topics covered include:

- “Logging on to the System” on page 41
- “Working with DataJoiner Instances” on page 41
- “Starting and Stopping DataJoiner” on page 42
- “Creating a DataJoiner Database” on page 43
- “Running Your Own Applications” on page 47
- “Running ODBC-Enabled Applications” on page 47

Logging on to the System

To use DataJoiner, log on with a valid user name that has the appropriate authority level for the commands that you want to execute.

You need administrative authority to start the database manager or to create the sample database. In other words, you must log on using a user name that is a member of the Administrators group. See “Creating a DataJoiner Administrator User Name” on page 32 for a description of valid user names.

For information on planning for and creating a secure database system, see the *DB2 DataJoiner Administration Supplement*.

Working with DataJoiner Instances

A DataJoiner instance is an environment where you store data and run applications. Depending on your needs, you might want to create more than one instance. For example, you can isolate a development environment from a production environment by defining a different database instance for each.

All of the files necessary to run the database instance are stored in a subdirectory with the same name as the instance name. This subdirectory is created in the directory where DataJoiner is installed. This directory is typically \SQLLIB.

A default instance (DB2) is created when you install DataJoiner for Windows NT. A Windows NT service also is created using the instance name. To create additional instances, use the **db2icrt** command. See “Creating a DataJoiner Administrator User Name” on page 32 for more information about naming rules and how to work with them.

Starting and Stopping DataJoiner

You must start the DataJoiner instance on the server before you can connect to a database, precompile an application, or bind a package to a database.

Starting and Stopping DataJoiner Using the Command Line

To start the DataJoiner instance, type the following command at a command prompt:

```
NET START instance_name
```

To stop the DataJoiner instance, type the following command at a command prompt:

```
NET STOP instance_name
```

If necessary, use the DB2 **FORCE** command to terminate all active applications that connect to the databases.

You can also start DataJoiner with the **db2start** command and stop DataJoiner with the **db2stop** command.

The **db2start** command causes DataJoiner to run only while you are logged on. With **NET START *instance_name***, DataJoiner runs as a service until explicitly stopped.

The same commands that are used to start and stop a DataJoiner instance are used to stop and start the DB2 Security Server. Substitute `db2ntseccserver` for *instance_name*.

Note to Informix users:

To use the **NET START** command on a system that accesses an Informix data source, you must run the Informix regcopy utility, which is found in the `%INFORMIXDIR%\bin` directory. The **db2start** command does not require you to run regcopy.

See the *DB2 Installation and Operation Guide* for more information about these commands.

Starting and Stopping DataJoiner Using Windows NT Tools

You can use the Windows NT Services window to control and configure DataJoiner services.

To start DataJoiner:

1. Access the Control Panel, either by double-clicking the **Program Manager** icon then the **Main** icon, or by selecting the **Start** button, then the **Settings** choice.
2. From the Control Panel, double-click the **Services** icon.
3. Select the **DB2-DB2** entry.
4. Click **Start**.

When the instance starts, the word "Started" appears in the **Status** column.

- If you want DataJoiner to start automatically after a reboot:
 - a. Click **Startup** to open a new window.
 - b. Select the **Automatic** radio button.
 - c. Click **OK**.
 - d. Click **Close**.
- If you prefer to always start DataJoiner manually, click **Close**.

To stop the DataJoiner instance from the Services window, select the **DB2 - DB2** entry, then click **Stop**.

Creating a DataJoiner Database

DataJoiner uses the catalog from a local database to maintain information about the DataJoiner heterogeneous environment. To create a DataJoiner database, use the **CREATE DATABASE** command.

The **CREATE DATABASE** command has the following syntax:

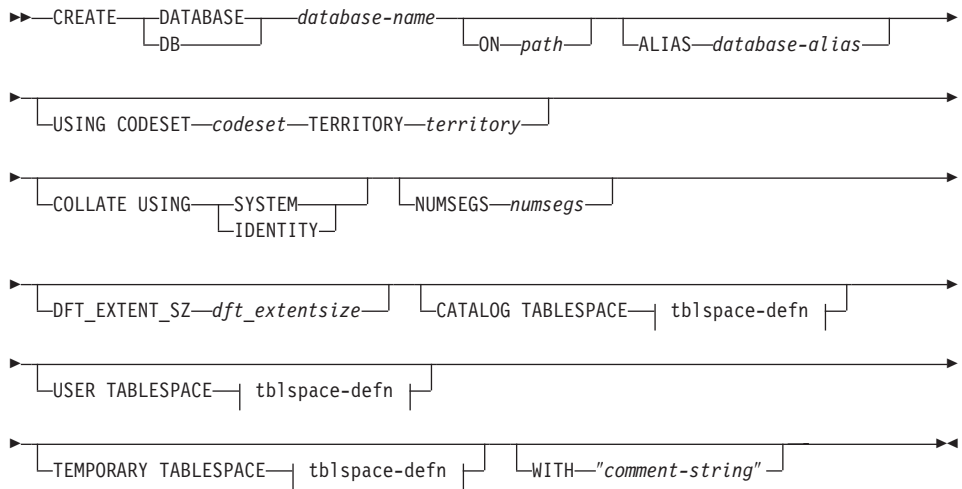


Figure 5. CREATE DATABASE Command Syntax

Notes about the CREATE DATABASE command:

- The code set and territory values that you specify must be a valid combination. For a list of valid combinations, see the *DB2 for Windows NT Installation and Operation Guide*.
- For details on the **tblspace-defn** parameters, see the CREATE TABLESPACE statement in the *DB2 SQL Reference*.

Parameters

database-name

Represents the name to be assigned to the database you are creating. This must be a unique name that differentiates the database from any other databases that exist within the DataJoiner instance. The name must conform to naming conventions for databases. Refer to the *DataJoiner Administration Supplement* for more information about naming conventions.

path

Specifies the path on which to create the database.

If you do not specify a path, the database is created in the default database path as specified in the database manager configuration file (dftdbpath parameter). To see the value of dftdbpath, use the **GET DATABASE MANAGER CONFIGURATION** command that is documented in the *DB2 Command Reference*. The default database path is the HOME directory of the owner of the DataJoiner instance you are using. The path can be a maximum of 215 characters.

database-alias

An alias for the database in the system database directory. If no alias is provided, the specified database name is used.

codeset and territory

In most cases, you do not need to specify these options.

If the code set and territory are not specified on the create database command, DataJoiner uses the territory and code set of the application that invokes the command. The application code page is derived from the active environment when the database connection is made.

If your database needs a code set and territory different from what is described, specify them in the **CREATE DATABASE** command. Examples of code set and territory are:

```
Database code set = IBM-850  
Database territory = En_US
```

For more information about code sets and territories that are supported by DataJoiner, see “Appendix D. National Language Support” on page 265.

COLLATE USING

Identifies the type of collating sequence to be used for the database. Possible collating sequence types are:

SYSTEM Indicates that the collating sequence is based on the current country code. This is the default.

IDENTITY Indicates that the collating sequence is the identity sequence, where strings are compared byte for byte.

When choosing the collating sequence, data usage and performance should be considered. However, when storing binary data (DB2 datatype CHAR(nn) FOR BIT), then **IDENTITY** should be used. You must use **IDENTITY** if the DataJoiner database will be used for replication. See the *DataJoiner Administration Supplement* for details on collating sequences.

numsegs and segpages

In most cases, you do not need to specify these options. The one exception is when you plan to store a large amount of user data in the local DataJoiner database.

For more information, see the *DB2 Administration Guide*.

dft_extentsize

Specifies the default extent size of table spaces in the database.

CATALOG TABLESPACE *tblspace-defn*

Specifies the definition of the table space that will hold the catalog tables, SYSCATSPACE. If not specified, SYSCATSPACE will be created as a

System Managed Space (SMS) table space with *numsegs* number of directories as containers, and with an extent size of *dft_extentsize*.

USER TABLESPACE *tblspace-defn*

Specifies the definition of the initial user table space, USERSPACE1. If not specified, USERSPACE1 will be created as an SMS table space with *numsegs* number of directories as containers, and with an extent size of *dft_extentsize*.

TEMPORARY TABLESPACE *tblspace-defn*

Specifies the definition of the initial user table space, TEMPSPACE1. If not specified, TEMPSPACE1 will be created as an SMS table space with *numsegs* number of directories as containers, and with an extent size of *dft_extentsize*.

comment-string

Describes the database or the database entry in the system directory. Do not use a carriage return or line feed character in the comment. Be sure to enclose the comment text in double quotes. The comment can be a maximum of 30 characters.

When you issue a **CREATE DATABASE** command, DataJoiner performs the following actions:

1. Creates the database.
2. Creates the system catalog tables and recovery log.
3. Catalogs the database in the local database directory on the path where the database was created and in the system database directory. If the database directories do not already exist, they are created. See the *DataJoiner Administration Supplement* for more information on database directories.

The database is cataloged with an alias that is the same as the database name. To assign another alias to the database, you can first uncatalog the database and then use the **CATALOG DATABASE** command, which is documented in the *DB2 Command and API Reference*.

4. Grants DBADM authority and CONNECT, CREATETAB, and BINDADD privileges to you as the database creator.
5. Assigns the code set and territory.
6. Calls the BIND executor to bind pre-defined DataJoiner bind files to the database.
7. Grants the BIND and EXECUTE privilege as PUBLIC for DataJoiner utilities, such as the Command Line Processor (CLP).

With DBADM authority, you can grant these privileges to (and revoke them from) other users or PUBLIC. If another administrator with SYSADM or DBADM authority over the databases revokes these privileges, you, as the database creator, retain your privileges.

When you create a database that will be accessed by most of your users, you might want to update the DB2DBDFT environment variable as described in “Chapter 21. Setting Environment Variables” on page 147. If so, set DB2DBDFT to the name of the newly created database.

For more information about the **CREATE DATABASE** command, see the *DB2 Command Reference*.

Running Your Own Applications

From a DB2 client, a database application program can access a database on a remote node without knowing the physical location of the database. The client determines the location of the database, manages the transmission of the requests to the database server, and returns the results.

To run a database client application that was developed using the Software Developer's Kit:

1. Ensure that the server is configured and running.
Be sure that the DataJoiner instance is started on the database server to which the application program is connecting. If it is not, issue a **NET START *instance_name*** command at the server.
2. Ensure that you know the security requirements on the database server. If you require a password or user name, you must have the correct ones.
3. Bind the utilities and the applications to the database. See “Step 4: Bind Packages used by DataJoiner and Other Applications” on page 77 for information about binding the utilities.
4. Run the application programs.

Running ODBC-Enabled Applications

Before you can use ODBC applications, such as Lotus Approach, to access data in DB2 databases, you must perform the following steps:

1. Verify that the IBM ODBC driver manager is installed:
 - a. Go to the **Control Panel** folder and double-click the **ODBC** icon. The Data Sources window opens.
 - b. Click **Drivers**.
 - c. Verify that **IBM DB2 ODBC Driver** appears in the list.
 - d. Close the window.
2. Catalog the database (and node if the database is remote). See the *DB2 Command Reference* for information about the **CATALOG DATABASE** and **CATALOG TCPIP NODE** commands.

3. Bind the ODBC driver bind files to the database. See “Step 4: Bind Packages used by DataJoiner and Other Applications” on page 77 for information.

The ODBC driver will autobind on the first connect to the database, provided that the user name has the appropriate privilege or authorization. The administrator might need to perform the first connect or explicitly bind the required files.

4. Use the DB2 Client Setup to register the database with the ODBC driver manager as a *data source*:
 - a. Open the **DB2 for Windows NT** folder.
 - b. Double-click the **DB2 Client Setup** icon. The DB2 Client Setup window opens.
 - c. Select **Assistance**. The DB2 Client Setup — Assistance window opens.
 - d. Select the **Allow programs access to a DB2 database via ODBC** push button.
 - e. Select a database to work with, then select **Next**. The Configure ODBC Data Source window opens.
 - f. Select **Data source and description** from the **Settings for** list. Enter a description for the data source and select **OK**.
 - g. Select **OK** to save the changes to the DB2CLI.INI file and to return to the Assistance window.
5. Start an application that uses ODBC. When you open ODBC databases from within the application, you are presented with a list of the databases that you can connect to, including the one that you just registered.

Part 3. Connecting DataJoiner to Data Sources

Chapter 7. Configuring Access to Data Sources

DataJoiner uses data access modules to communicate with data sources. A different data access module is required for each data source type.

About link-editing data access modules:

DataJoiner on UNIX requires a link-edit step that creates a data access module for a particular data source; however, DataJoiner for Windows NT provides pre-built data access modules. A link-edit step is not required.

Data access modules talk to a data source through the access methods that are provided by the data source for its clients. Therefore:

- A data source must be configured to accept clients through a specific access method.
- For some data sources, DataJoiner must use client software that is provided by the data source vendor.

Use Table 4 to determine which access methods apply to your installation.

Table 4. Data Sources and Their Access Methods

Data Source	Access Method	See...
DataJoiner V1	IBM DB2RA	page 67
DataJoiner V1.2 and V2	IBM DB2RA or DRDA	page 67 (for DB2RA) or page 57 (for DRDA)
DB2 Universal Database V5	IBM DB2RA or DRDA	page 67 (for DB2RA) or page 57 (for DRDA)
DB2 for OS/390, DB2/MVS, DB2 for OS/400, SQL/DS and DB2 for CS V2 ^{1,2,3}	DRDA	page 57
DB2 for CS (except DB2 for OS/2 V1 and DB2 Parallel Edition)	IBM DB2RA	page 67
DB2 Parallel Edition	IBM DB2RA or DRDA	page 67 (for DB2RA) or page 57 (for DRDA)
Informix OnLine V7.1 or higher (non-GLS, -XPS, and -SE versions only)	informix7 data access module	page 81
Microsoft SQL Server	ODBC	page 85
ODBC/XOpen CLI	Generic data access module	page 113

Table 4. Data Sources and Their Access Methods (continued)

Oracle	Oracle SQL*Net or Net8	page 91
Oracle RDB	ODBC	page 89
Sybase	Sybase Open Client	page 97
Sybase SQL Anywhere	ODBC	page 103
Teradata	Teradata data access module	page 105

Notes:

1. Although you can use DRDA to access DB2 for CS V2, IBM DB2RA is the recommended access method for DB2 for CS.
2. DRDA application server data sources that will be participating in two-phase commit transactions must use the DB2RA protocol; see “Configuring for Two-Phase Commit”.
3. DRDA3 application server data sources (DB2 for OS/390 and DB2 for OS/400 V4.2) can also use the drdaIP protocol. Two-phase commit is not supported for this protocol.

Loading Data Access Modules

You can control whether or not a data access module is loaded at DataJoiner initialization through the use of the DJXCOMM environment variable. Loading a data access module at initialization reduces system overhead by eliminating the need to load and unload a module each time a data source is accessed. See “Chapter 21. Setting Environment Variables” on page 147 for more information.

Configuring for Two-Phase Commit

This section shows you how to enable data sources to participate in two-phase commit transactions. Not all versions of all data sources supported by DataJoiner are capable of two-phase commit. Table 5 on page 53 shows which data sources support two-phase commit and lists additional requirements for certain data sources. The second column indicates if a data source, with the listed constraints, can participate in a two-phase commit transaction. The third column indicates if the data source supports the creation of new objects at the data source (using a pass-through session) while participating in a two-phase commit transaction. For more information about two-phase commit concepts and terminology, see the *DataJoiner Administration Supplement*.

Table 5. Data Sources Supporting Two-Phase Commit

Data Source	Supports two-phase commit?	Supports two-phase commit DDL?
Classic Connect	No	No
Cross Access	No	No
DB2 CS and DataJoiner, Version 1.2	Yes	Yes
DB2 CS and DataJoiner, Version 2	Yes	Yes
DB2 for OS/390, Version 3 and higher (with APARs PN67179 and PN70102)	Yes	Yes
DB2 for OS/400, before Version 3.1	No	No
DB2 for OS/400, Version 3.1 and higher	Yes	Yes
EDA/SQL	No	No
Generic Access API data sources	No	No
Informix Version 5	Yes	Yes
Informix, Versions 7.1 and 7.2, with TP/XA library	Yes	Yes
MS SQL Server (DBLIB)	No	No
MS SQL Server 4.2 (CTLIB) with XA library	No	No
MS SQL Server 6.5 (ODBC), using DataJoiner on NT	No	No
MS SQL Server 6.5 (ODBC), using DataJoiner V2 on UNIX operating systems	No	No
Oracle V7 and higher, without the XA library distributed database option	No	No
Oracle V7 and higher, with the XA library and the distributed database option on UNIX operating systems	Yes	No
Oracle V7.3 and higher, on Windows NT 3.51 and 4.0 operating systems (requires V7.3.3 Oracle Client at DataJoiner on NT operating systems)	Yes	No
Oracle RDB (all levels)	No	No
SQL Anywhere	No	No
SQL/DS, before Version 4.1	No	No
SQL/DS, Version 4.1 and higher	Yes	Yes
Sybase SQL Server (CTLIB) with XA library	Yes	No
Sybase SQL Server (DBLIB)	No	No
Teradata	No	No

Enabling Data Sources for Two-Phase Commit

Perform the following steps for data sources that will be participating in two-phase commits:

- For all data sources participating in two-phase commit transactions, define a transaction resolution password and ID for resynch operations. This is a special user name and password with the authority to COMMIT and ROLLBACK transactions. Then, using the CREATE USER MAPPING SQL statement, map this ID to the special ID, SYSTMDB.

DataJoiner will always use the SYSTMDB ID for the resynch agent that connects to RMs during resynch processing. This ID must be mapped to a remote user ID and password that has the authority to commit or rollback the transaction initiated on remote tables by users through DataJoiner.

For more information about CREATE USER MAPPING statements, see the *Application Programming and SQL Reference Supplement*.

- For non-DB2 data sources (Informix, Oracle, Sybase ctlib) install the required XA libraries for the data source.

For Sybase ctlib data sources, you must update the xa_config file to map Sybase node names to Logical Resource Manager (LRM) names. If a DataJoiner user has write authority for the xa_config file, this step is performed automatically by DataJoiner when the CREATE SERVER MAPPING statement is issued for the data source. If a user does not have write authority for the xa_config file, this step must be performed manually by the Sybase administrator.

- For DRDA server data sources, two-phase commit transactions require a DDCS gateway, which can be either standalone DDCS or DDCS as supplied with DataJoiner for AIX.

Restriction: Neither DDCS for Windows NT nor the DDCS functionality packaged with DataJoiner for Windows NT can be used as a DDCS gateway to DRDA server data sources using two-phase commit. Two-phase commit transactions from DataJoiner for Windows NT must go through a non-Windows NT DDCS gateway, such as DDCS for AIX or OS/2, or through DataJoiner for AIX.

There are three possible ways that you can use DDCS to access DRDA server data:

- Use an existing DDCS gateway that is installed as a standalone product. For customers who already have DDCS installed and configured, this is the simplest method.

- Use a separate DataJoiner instance that is installed as a DDCS gateway. This method allows you to use DataJoiner as a DDCS gateway that can be shared among multiple DataJoiner instances for transactions involving two-phase commit.
- Use the same DataJoiner instance as a DDCS gateway. When DataJoiner requests data from a DRDA server data source, that request is looped back to itself and handled by its DDCS gateway functionality. For customers who do not currently own standalone DDCS, this is the simplest method because it does not require a separate DataJoiner instance.

No matter which method you choose, the set up procedure for DDCS is the same. For information about configuring DDCS for two-phase commit, see the *DDCS for AIX Installation and Configuration Guide*.

Configuring DataJoiner for Two-Phase Commit

This section describes the steps you must perform at the DataJoiner instance for two-phase commit transactions.

- For non-DB2 data sources, check with the data source provider for instructions on installing XA libraries on the NT client where DataJoiner is installed.
- For DRDA server data sources, perform the following steps at the DataJoiner instance:

1. Catalog a TCP/IP node entry pointing from DataJoiner to the DDCS gateway; for example:

```
CATALOG TCPIP NODE ddcs_node_name REMOTE ddcs_machine_name
SERVER ddcs_port_name
```

where *ddcs_node_name* is a unique name for the DDCS gateway, *ddcs_machine_name* is the name of the system where the data source resides, and *ddcs_port_name* is the primary port name for use by data source clients.

2. Define the DDCS gateway to DataJoiner by issuing a CREATE SERVER MAPPING statement; for example:

```
CREATE SERVER MAPPING FROM server_name TO NODE "ddcs_node_name"
DATABASE "ddcs_dbname" TYPE DB2/MVS VERSION 3.1 PROTOCOL "db2ra"
```

Notice that db2ra is specified as the protocol. This is because DataJoiner is accessing a DDCS instance as opposed to accessing the DB2/MVS data source directly.

3. Enable each data source that will participate in two-phase commits by issuing a CREATE SERVER OPTION statement; for example:

```
CREATE SERVER OPTION two_phase_commit FOR SERVER server_name SETTING 'Y'
```

Alternately, you can modify the two-phase commit option at runtime by issuing the SET SERVER OPTION command.

See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about the mapping statements that are described in this section.

Overview of Configuration Steps

This section describes the steps that are required to configure DataJoiner to access data sources. The steps are similar for all data sources:

1. If necessary, install client software on the DataJoiner server.
2. Configure network communications.
3. Identify the node where the data source resides to DataJoiner.
4. Update DataJoiner's catalog views with information about the data source.
5. Test the connection to the data source.

DataJoiner can also be configured to access DB2 data sources directly, without the use of nicknames or pass-through sessions. The configuration steps for this method are described in detail in "Accessing DRDA Data Sources Using DataJoiner DDCS" on page 62 and "Accessing DB2RA Data Sources Outside the Heterogeneous Environment" on page 76.

Chapter 8. Accessing DRDA Data Sources

There are two ways to access data sources using DataJoiner's DRDA Application Requester (AR) functionality:

- DataJoiner provides access through the use of nicknames and the pass-through facility.
- DataJoiner also provides access for applications that use IBM's Distributed Database Connection Services (DDCS).

DRDA data sources can be accessed through SNA LU 6.2., while DRDA3 data sources (DB2 for OS/390 and DB2 for OS/400 V4.2) can be accessed through SNA LU 6.2 or TCP/IP. Configuration steps for both communications protocols are provided in "Chapter 19. Configuring Communications for DRDA Data Sources" on page 117.



A TaskGuide is available to help you easily configure DRDA data sources, or you can perform this step manually using the following instructions.

Accessing DRDA Data Sources Using Nicknames and Pass-through

This section demonstrates configuring DataJoiner to a DB2 for MVS data source. Other DRDA data sources are configured in a similar manner. For detailed information on other DRDA data sources, refer to the *DRDA Connectivity Guide* (SC26-4783).

The process of configuring DataJoiner and DRDA data sources for two-phase commit transactions is explained in "Configuring for Two-Phase Commit" on page 52.

The configuration steps that are required to access a DRDA data source using nicknames and the pass-through facility are:

1. Configure network communications
2. Update the node directory
3. Update system catalog views with server information and user information

Figure 6 on page 58 is an overview that shows how DataJoiner's DRDA application requester accesses a data source by using the catalog view, SYSCAT.SERVERS.

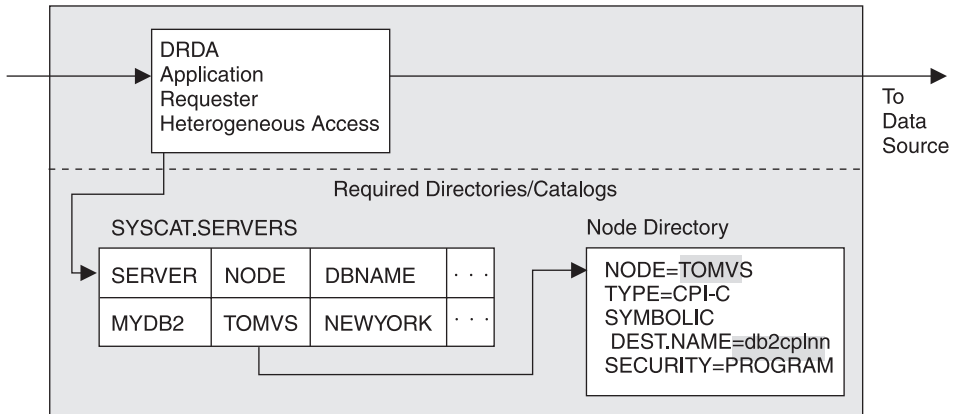


Figure 6. DataJoiner Application Requester

SNA users:

Figure 6 references the NODE directory entry, in this case TOMVS. The NODE directory then references the SNA Configuration Side Information Profile, db2cplnn. The connection is initiated from there.

The remainder of this section contains detailed instructions for the steps that are outlined above. It will be helpful to fill in the planning worksheets that are provided in “Appendix C. Worksheets for Use with DataJoiner DDCS” on page 261 as you configure your environment.

Step 1: Configure Network Communications

You can use the SNA LU 6.2 communications protocol to access DRDA data sources, and you can use either SNA LU 6.2 or TCP/IP to access DRDA3 data sources. See “Chapter 19. Configuring Communications for DRDA Data Sources” on page 117 for instructions.

Step 2: Update the Node Directory

Update the node directory by using either the DB2 command window that is located in the **DB2 for Windows NT** folder or the Database Director utility.

The following example shows the proper syntax for updating the node directory from a DB2 command window. Use the first command for APPC configurations; use the second command for TCP/IP configurations. In the first sample command, db2node is the name of the node in the node directory, and db2cpic is item 9 from “Windows NT Connection to Server Worksheet” on page 264:


```
DB2 CATALOG APPC NODE db2node REMOTE db2cpic SECURITY PROGRAM
```

or

```
DB2 CATALOG TCPIP NODE db2node REMOTE tcphost SERVER 446
```

where:

NODE

Is an arbitrary alias for the node that is being cataloged.

REMOTE

For APPC configurations is the symbolic destination name of the remote partner node. For TCP/IP configurations, REMOTE is the host name of the node where the target database resides.

SECURITY

Controls whether or not user names and passwords are passed to the DRDA Application Server (AS), which in this scenario is DB2/MVS. Valid values include PROGRAM, SAME, and NONE. The SECURITY parameter is used only for APPC configurations. See the *DataJoiner Administration Supplement* for more information.

SERVER

Can be either the service name of the database manager on the remote node, or the port number that is associated with that service name. The SERVER parameter is used only for TCP/IP configurations.

For more information about the **CATALOG...NODE** command, see the *DATABASE 2 Command Reference*.

Step 3: Update DataJoiner Catalog Views with Server and User Information

To access a data source, you must update the DataJoiner catalog by using DataJoiner DDL statements. This section explains how to do so by using the following DDL statements:

- CREATE SERVER MAPPING
- CREATE USER MAPPING
- CREATE SERVER OPTION
- CREATE NICKNAME

See the *DataJoiner Application Programming and SQL Reference Supplement* for more detailed information about DataJoiner DDL statements and when to use them.

CREATE SERVER MAPPING

Use the CREATE SERVER MAPPING DDL statement to update the SYSCAT.SERVERS view. This DDL statement is required to identify the DRDA data source to DataJoiner. The following guidelines apply when issuing the CREATE SERVER MAPPING statement:

- Choose a unique server name.
- Set NODE to the node you cataloged in “Step 2: Update the Node Directory” on page 58. This value is case sensitive.
- Set DATABASE to the name of a database at the DB2 server that you are mapping. For a DB2/MVS or DB2 for OS/390 data source, it is the LOCATION name in the DB2/MVS Bootstrap data set (BSDS) of the DB2 subsystem. This value is case sensitive.
- Set TYPE to the data source type you are mapping.
- Set VERSION to the correct version of the data source.
- Set PROTOCOL to "drda" if using APPC, or to "drdaIP" if using TCP/IP. This value is case sensitive and must be enclosed in delimiters.
- Set AUTHID to an authorization name at the data source that has SYSADM or equivalent authority. This value will be transformed to uppercase unless enclosed in quotation marks.
- Set PASSWORD to the password that is associated with the AUTHID. This value will be transformed to uppercase unless enclosed in quotation marks.

A sample statement for a DB2/MVS data source that is accessed using APPC is:

```
CREATE SERVER MAPPING FROM db2mvs TO NODE "db2node"  
DATABASE "db2mvs01" TYPE db2/mvs VERSION 3.1 PROTOCOL "drda"  
AUTHID sysadm PASSWORD db2mvsadm
```

A sample statement for a DB2/MVS data source that is accessed using TCP/IP is:

```
CREATE SERVER MAPPING FROM st1714a TO NODE "db2node"  
DATABASE "st1714a" TYPE db2/mvs VERSION 5.1 PROTOCOL "drdaIP"  
AUTHID sysadm PASSWORD admin
```

Attention:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"drda\".

CREATE USER MAPPING

Use the CREATE USER MAPPING DDL statement to update the SYSCAT.REMOTEUSERS view. This DDL statement is used to map information about authorization IDs and passwords that are used to access a data source. The following example shows the proper syntax of the CREATE USER MAPPING DDL statement:

```
CREATE USER MAPPING FROM benham TO SERVER db2mvsa
AUTHID SMITH PASSWORD JONES
```

where:

- benham is a local DataJoiner authorization name.
- db2mvsa is the name of the server that contains the remote authorization ID, which, in this example, is SMITH.
- JONES is the password that is associated with the remote authorization ID.

CREATE SERVER OPTION

Use the CREATE SERVER OPTION DDL statement to update the SYSCAT.SERVER_OPTIONS view. This DDL statement is used to customize your DataJoiner configuration. The following example shows the proper syntax of the CREATE SERVER OPTION DDL statement for the pushdown option.

```
CREATE SERVER OPTION pushdown FOR SERVER db2mvsa
SETTING 'Y'
```

where:

- pushdown is the option that controls whether or not DataJoiner will allow remote data sources to evaluate operations locally.
- db2mvsa is the name of the server for which this option is being implemented.
- 'Y' indicates that the option is allowed.

CREATE NICKNAME

Use the CREATE NICKNAME DDL statement to map the server name, remote authorization ID, and table name of a remote data source into a single nickname.

Attention:

Before creating nicknames, you should understand the considerations and restrictions associated with them as documented in the *DataJoiner Application Programming and SQL Reference Supplement*.

The following example shows the proper syntax of the CREATE NICKNAME DDL statement:

```
CREATE NICKNAME deptx.proj_info FOR db2mvs.a.SMITH.proj
```

where:

- deptx is an optional schema or qualifier. If none is specified, this value defaults to the user ID of the creator.
- proj_info is an arbitrary identifier.
- db2mvs.a is the name of the server where the remote view or table is located.
- SMITH is the schema to which the view or table belongs.
- proj is a remote view or table name.

Accessing DRDA Data Sources Using DataJoiner DDCS

Applications that use IBM's Distributed Database Connection Services (DDCS) to access DRDA data sources can also be used with DataJoiner's DRDA application requester (AR) functionality. There are two reasons you might want to use this method:

- So applications can connect directly to DRDA databases without using DataJoiner nicknames or pass-through sessions.
- As an alternative means of binding the packages that are used by DataJoiner to these data sources (DataJoiner also binds these packages when a CREATE SERVER MAPPING statement is issued for a DRDA data source).

Use the following steps to access DRDA data sources using DataJoiner's AR functionality:

1. Configure network communications
2. Update the node directory
3. Catalog a DCS directory entry
4. Catalog a database directory entry
5. Bind packages that are used by DataJoiner and other applications

Step 1: Configure Network Communications

See "Chapter 19. Configuring Communications for DRDA Data Sources" on page 117 for instructions.

Step 2: Update the Node Directory

Update the node directory by using either the DB2 command window that is located in the **DB2 for Windows NT** folder, or the Database Director utility.

The following example shows the proper syntax for updating the node directory from a DB2 command window. Use the first command for APPC configurations; use the second command for TCP/IP configurations. In the first sample command, db2node is the name of the node in the node directory, and db2cpic is item 9 from “Windows NT Connection to Server Worksheet” on page 264:

```
CATALOG APPC NODE db2node REMOTE db2cpic SECURITY PROGRAM
```

or

```
CATALOG TCPIP NODE db2node REMOTE tcphost SERVER db2srvr
```

where:

NODE

Is an arbitrary alias for the node that is being cataloged.

REMOTE

For APPC configurations is the symbolic destination name of the remote partner node. For TCP/IP configurations, REMOTE is the host name of the node where the target database resides.

SECURITY

Controls whether or not user names and passwords are passed to the DRDA AS. Valid values include PROGRAM, SAME, and NONE. The SECURITY parameter is used only for APPC configurations. See the *DataJoiner Administration Supplement* for more information.

SERVER

Can be either the service name of the database manager on the remote node, or the port number that is associated with that service name. The SERVER parameter is used only for TCP/IP configurations.

For more information about the **CATALOG APPC NODE** or **CATALOG TCPIP NODE** command, see the *DATABASE 2 Command Reference*.

Step 3: Catalog a DCS Directory Entry

Catalog an entry in the DCS directory for the database; for example:

```
CATALOG DCS DATABASE db2db AS new_york
```

where:

- db2db is the alias of the target database being cataloged.
- new_york is the target database name.

For more information about the **CATALOG DCS DATABASE** command, see the *DATABASE 2 Command Reference*.

Step 4: Catalog a Database Directory Entry

Update the system database directory; for example:

```
CATALOG DATABASE db2db AS nyc3 AT NODE db2node AUTHENTICATION DCS
```

where:

- db2db is the database name that was specified in the **CATALOG DCS DATABASE** command in the previous step.
- nyc3 is the database alias.
- db2node is the name of the node from “Step 2: Update the Node Directory” on page 62.

For more information about the **CATALOG DATABASE** command, see the *DATABASE 2 Command Reference*.

Step 5: Bind Packages Used by DataJoiner and Other Applications

Before binding:

If the database manager is already active, stop and restart the DataJoiner instance to activate the changes you have just made. Use the following commands:

```
NET STOP DB2  
NET START DB2
```

DB2 is the default instance name. If you are using a different instance name, use your own value instead.

To simplify the bind procedure, bind files are grouped together in different .lst files, each one being specific to a type of server. You must first connect to the database, then issue the **bind** command against the database using the appropriate .lst file.

To bind the necessary packages for the DB2 Client Application Enabler products in use at the DRDA server, issue the following commands:

```
CONNECT TO dbname USER userid USING password  
BIND <path>/@<DRDA_server>.lst BLOCKING ALL  
SQLERROR CONTINUE MESSAGES bind.msg GRANT PUBLIC  
CONNECT RESET
```

where *dbname* is the database name; the *userid/password* pair are used for authentication; <DRDA_server>.lst is the appropriate .lst file for a specific DRDA server, as shown in the following table:

DRDA Server	<DRDA_server>.lst
MVS	ddcsmvs.lst

DRDA Server	<DRDA_server>.lst
VSE	ddcsvse.lst
VM	ddcsvm.lst
OS/400	ddcs400.lst

The following example shows a typical command sequence for binding packages at a DB2/MVS server:

```
connect to db2mvs
bind c:\sqllib\bnd\@ddcsmvs.lst blocking all grant public
bind c:\sqllib\bnd\@applycs.lst isolation cs blocking all grant public
bind c:\sqllib\bnd\@applyur.lst isolation ur blocking all grant public
```

You will also need to bind your clients to the DataJoiner database. Use the `db2ubind.lst` and the `db2cli.lst` files, which are contained in `/usr/lpp/djx_02_01/bnd:`

```
CONNECT TO dbname USER userid USING password
BIND <path>/@db2ubind.lst MESSAGES bind.msg GRANT PUBLIC
BIND <path>/@db2cli.lst MESSAGES clibind.msg GRANT PUBLIC
```

where:

- *dbname* is the name of the database to which you wish to connect.
- *userid/password* are used for authentication.
- *<path>* is the full path name of the directory where the bind files are located, such as `\SQLLIB\BND`.
- `bind.msg` and `clibind.msg` are the output message files.
- **GRANT PUBLIC** grants EXECUTE and BIND privileges to PUBLIC.

Notes about binding:

- You must run the **BIND** command separately for each database that you wish to access.
- The `db2ubind.lst` file contains the list of bind (.bnd) files that are required to create the packages for the database utilities. The `db2cli.lst` file contains the list of bind (.bnd) files that are required to create packages for the DB2 CLI and the DB2 ODBC driver.
- Binding may take a few minutes to complete.

If you configured the DRDA AR just to bind DataJoiner packages to the DRDA database, then the DRDA data source can be uncataloged after you finish binding. However, the DRDA database will need to remain in the DataJoiner database directory if the DRDA database will be accessed directly by applications such as DataJoiner Replication and DataJoiner Replication Administration. For more information about binding DRDA data sources, see *Installing and Using DB2 Clients for Windows 95 and Windows NT* and the *IBM DDCS User's Guide* (SC09-1923-00).

Chapter 9. Accessing DB2RA Data Sources

DB2RA refers to the DB2 format and protocol between clients and servers. It is proprietary and equivalent to Oracle's SQL*Net and Sybase's Open Client/Open Server protocols, which are also proprietary. DataJoiner uses DB2RA natively to communicate between clients and DataJoiner, and between DataJoiner and DB2/CS and Universal Database data sources.

There are two ways to access DB2RA data sources:

- DataJoiner provides access through the use of nicknames and the pass-through facility.
- DataJoiner also provides direct access to DB2RA data sources without using nicknames or pass-through sessions.

Supported DB2RA data sources must have client support capabilities installed. See the data source documentation for information about configuring the data source.

DataJoiner supports TCP/IP, IPX/SPX, and APPC. Configuring DataJoiner to use any of these protocols to access a DB2RA data source is similar to configuring a client (CAE) to access DataJoiner.



A TaskGuide is available to help you easily configure DB2RA data sources, or you can perform this step manually using the following instructions.

Accessing DB2RA Data Sources Using Nicknames and Pass-through

The configuration steps that are required to access a DB2RA data source through the use of nicknames and the pass-through facility are:

1. Configure network communications
2. Update the node directory
3. Update system catalog views with server information and user information

Step 1: Configure Network Communications

You can use the following communications protocols to access DB2RA data sources:

- APPC; see "Configuring DataJoiner DDCS Using SNA" on page 118
- TCP/IP; see "Configuring TCP/IP for DB2RA Data Sources" on page 68

- IPX/SPX; see “Configuring IPX/SPX for DB2RA Data Sources” on page 70

Configuring TCP/IP for DB2RA Data Sources

This section shows how to use TCP/IP to establish a connection from DataJoiner to a DB2/CS data source. This section can be used as a guide for configuring any DB2RA data source that is being accessed via TCP/IP. The instructions here are similar to those used to configure a client to DataJoiner. The difference is that here DataJoiner is a client to DB2 instead of CAE or SDK being a client to DataJoiner.



Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem on one host might not be notified of the failure of its partner on a different host. As a result, a DataJoiner server accessing a data source using TCP/IP, or the corresponding process at the data source, might sometimes appear to be hung. You can use two network options, `tcp_keepidle` and `tcp_keepintvl`, to tune a specified time interval before the failure is detected.

The values for these options are system wide and apply to all TCP/IP socket applications for which the `SO_KEEPALIVE` socket option is enabled, not just DataJoiner.

To set or display the network options, use the `no` command.

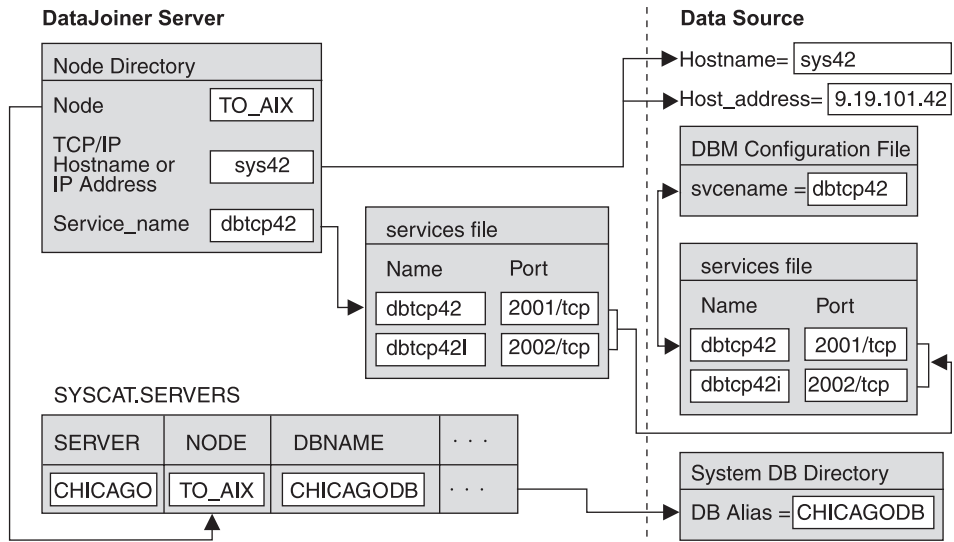


Figure 7. DB2RA Data Source via TCP/IP

The values that are used in this section are in Figure 7 on page 68. We also provide a blank template in Figure 8 on page 70. You can fill in the blank template with your system's information and then complete the instructions in this section by plugging your own values in where appropriate.

On the data source workstation:

1. Create the database if you did not already. You will need this database's name or its alias to be able to complete the configuration that is needed on DataJoiner.
2. Make sure that each DB2RA data source is installed and configured so it enables communication support for clients over TCP/IP. If this was not already done, see the data source's installation guide for instructions on how to do so.

You need two additional pieces of TCP/IP-related information before you can complete these instructions:

- The TCP/IP port numbers for the data source as they are stored in the TCP/IP services file on the data source's system. In the example, the entries would look like this:

```
dbtcp42      2001/tcp      # Data source primary connect port
dbtcp42i    2002/tcp      # Data source interrupt port
```

Exception:

Only Version 1 clients need the interrupt port number.

- The host name of the system where the data source resides. To get this name, issue the **hostname** command from a command prompt on the data source's system. In the example, the hostname is sys42.

On the DataJoiner server workstation:

1. Update the TCP/IP services file that is typically found in C:\WINNT\system32\drivers\etc with the port numbers. All DB2RA data sources communicate with TCP/IP clients through ports that are defined in the services file. If DataJoiner is installed on the same system as the data source, the ports are probably already defined as needed. If DataJoiner is installed on a different system than the data source, find the data source's port numbers in its system's services file. In the example, the entries would look like this:

```
dbtcp42      2001/tcp      # Data source primary connect port
dbtcp42i    2002/tcp      # Data source interrupt port
```

Alternately, you can specify the port numbers on the server parameter of the **CATALOG TCPIP NODE** command as discussed in "Step 2: Update the Node Directory" on page 73.

2. As an administrator, update the TCP/IP hosts file for the data source workstation, if necessary.

Whether you must update the hosts file depends on how TCP/IP is configured on your network. Part of the network must translate the remote hostname that is specified on the **CATALOG TCPIP NODE** command (sys42 in the example) to an address. If your network has a name server that recognizes the host name, you do not need to update the hosts file. Otherwise, you need an entry in the hosts file for the remote host name. See your network administrator to learn how your network is configured.

Use the following template to record your TCP/IP configuration values.

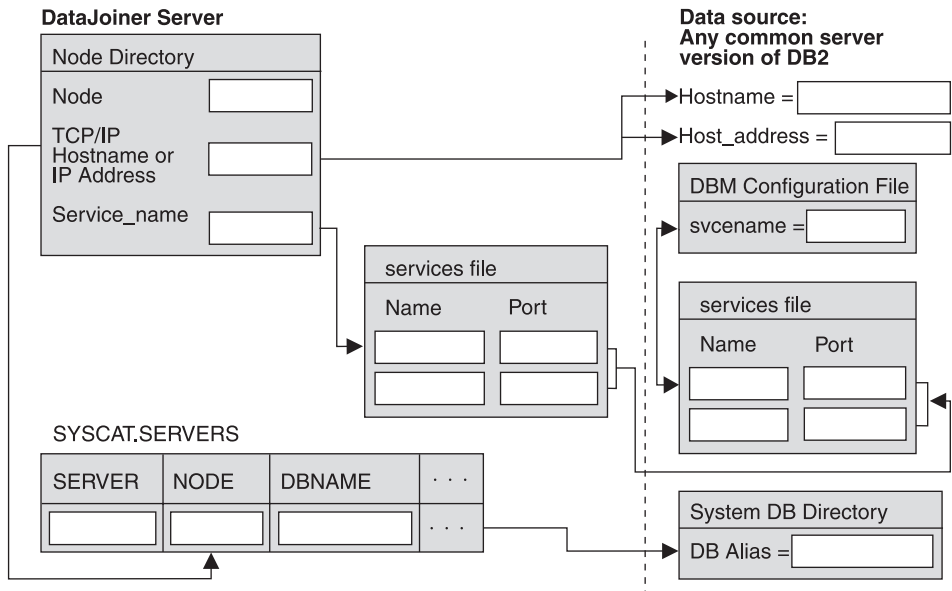


Figure 8. DB2RA Data Source Planning Template for TCP/IP

Configuring IPX/SPX for DB2RA Data Sources

To enable IPX/SPX support for DataJoiner server you must:

- Participate in a Novell NetWare LAN environment.
- Update the database manager configuration file.
- Register the DataJoiner server at the NetWare file server (file server addressing only).
- Set the DB2COMM environment variable.
- Start the database manager (**db2start**).

Configuring IPX/SPX for the Server: IPX/SPX support is part of the base Windows NT operating system. Ensure that you have installed the NWLink IPX/SPX network software and have a Novell NetWare LAN environment.

You can access a DataJoiner server through IPX/SPX as follows:

- **File Server Addressing.**

The client connects to the DataJoiner server by retrieving the server's address from a NetWare file server.

- **Direct Addressing.**

The client connects to the DataJoiner server by directly specifying the NetWare internetwork address of the server. The NetWare file server is bypassed. No NetWare file server needs to reside on the network with the client and server.

The NetWare information you need to update the database manager configuration file is as follows:

Fileserver

Specifies the name of the NetWare fileserver where the internetwork address of the database manager will be registered.

The internetwork address of the database manager is stored in the bindery at the NetWare file server.

Object name

Represents a particular database manager in the network, and should be unique for each DataJoiner server instance that is registered at a NetWare file server.

Socket number

Specifies a well-known static socket number and represents the connection endpoint in a DataJoiner server's internetwork address. It must be entered in the database manager configuration file in hexadecimal.

The socket number defaults to 0x879E (change this value if running more than 1 DB2 server instance). DataJoiner has registered well-known sockets with Novell in the range 0x879E to 0x87A2 for the customer's use. If you run more than 5 instances on the server machine, you must prevent socket collisions for instances 6 and higher by choosing a socket number that is NOT 0x0000 or in the dynamic socket range 0x4000 to 0x7FFF.

The number 0x0000 is reserved for allocating dynamic sockets, and the range of numbers from 0x4000 to 0x7FFF is used for dynamic sockets.

The socket number (*ipx_socket*) must be unique for every DataJoiner server instance on a machine. It must also be unique among all IPX/SPX applications running on the DataJoiner server machine to ensure that the DataJoiner server is able to listen to incoming IPX/SPX connections that are using this socket number.

Updating the Database Manager Configuration File for IPX/SPX: As described above, the database manager configuration file at the server contains the following parameters that are specific for IPX/SPX support:

- *fileserver*
- *objectname*
- *ipx_socket*

File Server Addressing: Configure this type of addressing if the DataJoiner server will be accessed by clients using:

- File server addressing and direct addressing
- or
- File server addressing only.

Restriction:

For file server addressing, the following characters are not valid for the *fileserver* or *objectname* parameters: / \ : ; , * ?

These configuration parameters can be entered or modified in the database manager configuration file by using one of the following tools:

- The Command Line Processor, documented in the *DB2 Command Reference*
For example:

```
db2 update database manager configuration using fileserver netwsrv
db2 update database manager configuration using objectname db2inst1
db2 update database manager configuration using ipx_socket 879F
```
- The Configuration API, documented in the *DB2 API Reference*
- The DB2 Database Director

Direct Addressing: Configure this type of addressing if the DataJoiner server will be accessed by clients using only direct addressing.

If you are using direct addressing only, you may specify the *fileserver* and *object name* parameters as '*', for example:

```
db2 update database manager configuration using fileserver * objectname *
```

To make this change to the configuration file effective, you must stop and start the database manager.

Registering the DataJoiner Server: It is not necessary to register the DataJoiner server if you use direct addressing only.

Once the IPX/SPX parameters have been configured, the administrator must register the DataJoiner server in the bindery at the NetWare file server before starting the database manager as follows:

```
db2 register db2 server in nwbindery user userid password password
```

where *userid* and *password* are your Novell file server login user ID and password which must have SUPERVISOR or Workgroup Manager security equivalence.



Hint for NetWare setup:

When registering at a NetWare 4.x file server (which uses directory services and provides bindery emulation capability), the *userid* used to register/deregister must be created with the same context as the "current bindery context that is used by Directory Services when it does bindery emulation" (as well as having supervisory or equivalent authority).

You can find the bindery emulation context currently in use by checking the bindery emulation setting on the NetWare 4.x file server (with the *SERVMAN* utility).

Notes about registering the DataJoiner server:

- The **REGISTER** command must be issued locally from a DataJoiner server (it is not supported remotely) *ONCE* before the first **db2start** for each DataJoiner server instance.
- If you move the DataJoiner server on your network, change its address, or if other IPX/SPX configuration changes are necessary, you must *FIRST* deregister the DataJoiner server in the bindery at the NetWare file server, make the change, then re-register the DataJoiner server.

Setting DB2COMM for IPX/SPX: You must set a value of *ipxspx* in the DB2COMM environment variable to enable IPX/SPX support when a **db2start** command is issued.

Step 2: Update the Node Directory

Catalog a node entry that points to the location of the data source. For example, if using TCP/IP you would issue the following command:

```
CATALOG TCPIP NODE db2node REMOTE sys42 SERVER dbtcp42
```

where:

- *db2node* is an arbitrary alias for the node being cataloged.

- *sys42* is the host name of the system where the data source resides.
- *dbtcp42* is the primary port name that is defined by the data source for use by data source clients.

For more information about the **CATALOG...NODE** command, see the *DATABASE 2 Command Reference*.

Step 3: Update DataJoiner Catalog Views with Server and User Information

To access a DB2RA data source, you must update the DataJoiner catalog by using DataJoiner DDL statements. This section explains how to do so by using the following DDL statements:

- CREATE SERVER MAPPING
- CREATE USER MAPPING
- CREATE SERVER OPTION
- CREATE NICKNAME

See the *DataJoiner Application Programming and SQL Reference Supplement* for more detailed information about DataJoiner DDL statements and when to use them.

CREATE SERVER MAPPING

Use the CREATE SERVER MAPPING DDL statement to update the SYSCAT.SERVERS view. This DDL statement is required to identify the DB2RA data source to DataJoiner. The following guidelines apply when issuing the CREATE SERVER MAPPING statement:

- Choose a unique server name.
- Set **NODE** to the node you cataloged in “Step 2: Update the Node Directory” on page 73. This value is case sensitive.
- Set **DATABASE** to the name of a database at the server that you are mapping. This value is case sensitive.
- Set **TYPE** to the data source type you are mapping.
- Set **VERSION** to the correct version of the data source.
- Set **PROTOCOL** to “db2ra”. This value is case sensitive and must be enclosed in delimiters.
- Set **AUTHID** to an authorization name at the data source that has SYSADM or equivalent authority. This value will be transformed to uppercase unless enclosed in quotation marks.
- Set **PASSWORD** to the password that is associated with the AUTHID. This value will be transformed to uppercase unless enclosed in quotation marks.

A sample statement for a DB2/CS data source is:

```
CREATE SERVER MAPPING FROM db2v2cs1 TO NODE "db2node"  
DATABASE "invoices" TYPE DB2/CS VERSION 2.1 PROTOCOL "db2ra"  
AUTHID sysadm PASSWORD admin
```

Attention:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"db2ra\".

CREATE USER MAPPING

Use the CREATE USER MAPPING DDL statement to update the SYSCAT.REMOTEUSERS view. This DDL statement is used to map information about authorization IDs and passwords that are used to access a data source. The following example shows the proper syntax of the CREATE USER MAPPING DDL statement:

```
CREATE USER MAPPING FROM benham TO SERVER db2csa  
AUTHID SMITH PASSWORD JONES
```

where:

- benham is a local DataJoiner authorization name.
- db2csa is the name of the server that contains the remote authorization ID, which, in this example, is SMITH.
- JONES is the password associated with the remote authorization ID.

CREATE SERVER OPTION

Use the CREATE SERVER OPTION DDL statement to update the SYSCAT.SERVER_OPTIONS view. This DDL statement is used to customize your DataJoiner configuration. The following example shows the proper syntax of the CREATE SERVER OPTION DDL statement for the pushdown option.

```
CREATE SERVER OPTION pushdown FOR SERVER db2csa  
SETTING 'Y'
```

where:

- pushdown is the option that controls whether or not DataJoiner will allow remote data sources to evaluate operations locally.
- db2csa is the name of the server for which this option is being implemented.

- 'Y' indicates that the option is allowed.

CREATE NICKNAME

Use the CREATE NICKNAME DDL statement to map the server name, remote authorization ID, and table name of a remote data source into a single nickname.

Attention:

Before creating nicknames, you should understand the considerations and restrictions associated with them as documented in the *DataJoiner Application Programming and SQL Reference Supplement*.

The following example shows the proper syntax of the CREATE NICKNAME DDL statement:

```
CREATE NICKNAME deptx.proj_info FOR db2csa.SMITH.proj
```

where:

- deptx is an optional schema or qualifier. If none is specified, this value defaults to the user ID of the creator.
- proj_info is an arbitrary identifier.
- db2csa is the name of the server where the remote view or table is located.
- SMITH is the schema to which the view or table belongs.
- proj is a remote view or table name.

Accessing DB2RA Data Sources Outside the Heterogeneous Environment

The steps that are required to configure DataJoiner to access a DB2RA data source without using nicknames or pass-through sessions are:

1. Configure network communications
2. Update the node directory
3. Catalog a database
4. Bind packages used by DataJoiner and other applications

Step 1: Configure Network Communications

You can use the following communications protocols to access DB2RA data sources:

- APPC; see “Configuring DataJoiner DDCS Using SNA” on page 118
- TCP/IP; see “Configuring TCP/IP for DB2RA Data Sources” on page 68
- IPX/SPX; see “Configuring IPX/SPX for DB2RA Data Sources” on page 70

Step 2: Update the Node Directory

Catalog a node entry that points to the location of the data source. For example, if using TCP/IP you would issue the following command:

```
CATALOG TCPIP NODE db2node REMOTE sys42 SERVER dbtcp42
```

where:

- db2node is an arbitrary alias for the node being cataloged.
- sys42 is the host name of the system where the data source resides.
- dbtcp42 is the primary port name that is defined by the data source for use by data source clients.

For a complete discussion of the **CATALOG...NODE** command, see the *DATABASE 2 Command Reference*.

Step 3: Catalog a Database Directory Entry

Catalog a database directory entry for the database; for example:

```
CATALOG DATABASE db2db AS nyc3 AT NODE db2node AUTHENTICATION server
```

where:

- db2db is the name of the database entry in the system database directory.
- nyc3 is the database alias.
- db2node is the name of the node from “Step 2: Update the Node Directory”.

For a complete discussion of the **CATALOG DATABASE** command, see the *DATABASE 2 Command Reference*.

Step 4: Bind Packages used by DataJoiner and Other Applications

A *package* is an object that includes all the information that is needed to process specific SQL statements from a single source file. Some database utilities, such as IMPORT, EXPORT, REORG, the command line processor, call level interface, and the binder program, have associated bind files that create packages at the database. These packages must be bound to the database before the utilities can be used with the database.

The bind files are grouped together in different .lst files in the \SQLLIB\BND directory, each one being specific to a server. You may bind the database packages to your databases by using either the DB2 Client Setup or command line processor commands.

Using the DB2 Client Setup

The following steps show you how to bind database components to the sample database:

1. Open the **DB2 for Windows NT** folder.
2. Double-click the **DB2 Client Setup** icon. The DB2 Client Setup window opens.
3. Select **Assistance**. The DB2 Client Setup - Assistance window opens.
4. Select **Bind Program to Database**.
5. Select a database with which to work. Select **Next** to continue.
6. Specify the bind files to use by filling in the Bind window. To bind the utilities, DB2 CLI, and ODBC to the SAMPLE database, using the DB2UBIND.LST and DB2CLI.LST files that are located in the E:\SQLLIB\BND directory, fill in the following information:

Type of file	Bind and list file
Drive	E:
Directory	\SQLLIB\BND
File	DB2UBIND.LST, DB2CLI.LST

7. Select **Bind** to continue.
8. Enter your **user name** and **password** in the next window. Select **OK** to bind the files.
9. A progress window appears providing any information or error messages that arise.
10. Select **Close** to return to the Assistance window.
11. See the notes at the end of this section to determine if any are applicable to your environment.

Using the Command Window

Change directory to the E:\SQLLIB\BND directory and issue the following commands:

```
CONNECT TO dbname
BIND @db2ubind.lst MESSAGES bind.msg GRANT PUBLIC
BIND @db2cli.lst MESSAGES clibind.msg GRANT PUBLIC
CONNECT RESET
```

where:

- *dbname* is the name of the database to which you wish to connect.
- *bind.msg* and *clibind.msg* are the output message files.
- GRANT PUBLIC grants EXECUTE and BIND privileges to all users.

Notes about binding:

- You must run the **BIND** command separately for each database that you want to access.
- In a network environment, if you are using clients that run on different operating systems or are at different versions, then you must bind the utilities once for each operating system / version combination.
- If the utilities are bound with GRANT PUBLIC, they need to be bound to each database only once for each operating system / version combination.
- The DB2UBIND.LST file contains the list of bind (.BND) files required to create the packages for the database utilities. The DB2CLI.LST file contains the list of bind (.BND) files that are required to create packages for the call level interface and the DB2 ODBC driver.
- Binding may take a few minutes to complete.

Chapter 10. Accessing Informix Data Sources

DataJoiner for Windows NT supports Informix OnLine Version 7.2 or higher.

The following section assumes that the Informix server is installed on a different machine than DataJoiner and the Informix client code. If the Informix server and DataJoiner are installed on the same machine, follow the same instructions but do not set the SQLRM environment variable.

Requirements for Informix transaction logging:

To maintain transaction integrity, DataJoiner requires that transaction logging be enabled for your Informix sources. If transaction logging is disabled, DataJoiner is unable to rollback units of work.

Using the Informix Version 7 Data Access Module

To configure DataJoiner access to Informix databases through Informix Version 7.2:

1. If you have not already done so, install the Informix esql/rt or equivalent libraries on the DataJoiner system.
2. Use the Informix - Setnet 32 tool to configure **Server Information** and **Host Information**.

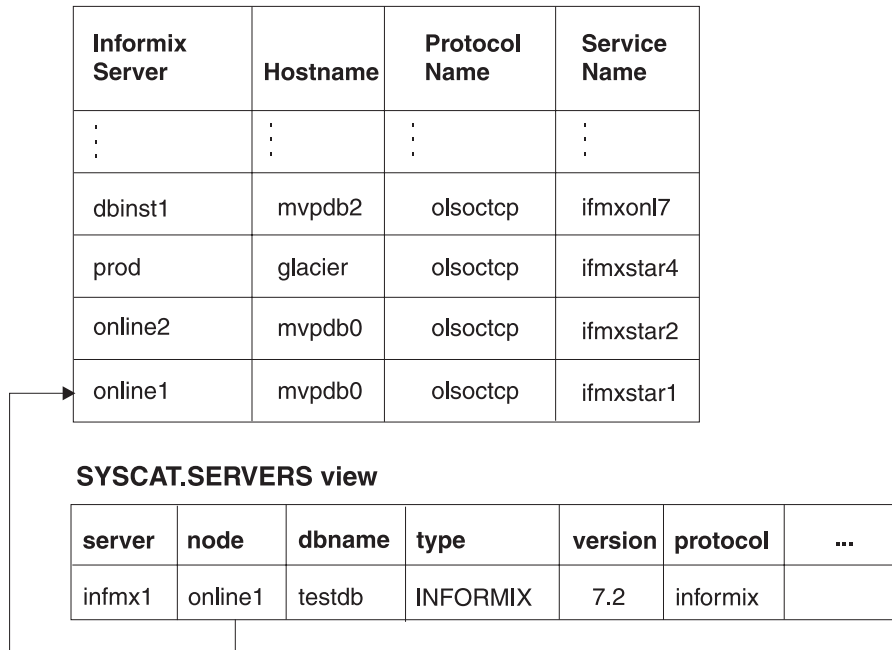


Figure 9. Informix V7 Data Access Module

3. Add the service name and port number for the Informix data source to the TCP/IP services file that is typically found in C:\WINNT\system32\drivers\etc.
4. Use the CREATE SERVER MAPPING DDL statement to define each Informix database you expect to access via Informix Version 7. Specify the same protocol for each database. When performing server mapping:
 - Choose a unique server name.
 - Set NODE to the appropriate Informix dserver name. This value is case sensitive.
Note that you do not need to use the **CATALOG NODE** command for the data sources that are accessed through the Informix Version 7 data access module.
 - Set DATABASE to the name of an Informix database at the Informix server that is listed in the NODE column. This value is case-sensitive.
 - Set TYPE to informix.
 - Set VERSION to the version of Informix that the Informix server is running. For example, 7.2 for Informix 7.2.
 - Set PROTOCOL to "informix7". This value is case sensitive.

A sample entry for Informix Version 7 is:


```
CREATE SERVER MAPPING FROM infrmxb TO NODE "infrmxb1" DATABASE "test_db"  
TYPE informix VERSION 7.2 PROTOCOL "informix7"
```

Preserving case-sensitivity:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \
"informix7\".

5. Update the environment of the DataJoiner instance to include the following lines:

```
INFORMIXDIR="informix-home-directory"  
INFORMIXSERVER="dbserver-name"
```

Where *informix-home-directory* is the directory in which you installed the e/sql library in step 1 on page 81, and *dbserver-name* is the name that is entered in the first column of the sqlhosts file.

Some releases of Informix require you to set INFORMIXSERVER with the name of a default server. However, this value is not used by DataJoiner. Instead, DataJoiner uses the dbserver-name set with the NODE parameter of the CREATE SERVER MAPPING statement.

See the *Informix—OnLine Administrator's Guide* for more information on these environment variables.

6. If you want the Informix data access module to start dynamically when the DataJoiner instance is started, update the environment of the DataJoiner instance to include an entry for the DJXCOMM variable. The name of the Informix7 data access module should be specified. If you do not set the DJXCOMM variable, you will need to manually start the Informix data access module.

For example, if you are using a data access module that is named informix7, the entry would be:

```
DJXCOMM=informix7
```

Ensure that there are no spaces on either side of the equal sign (=). After changing environment variables, you need to log off the system and log back on to activate your changes. See “Chapter 21. Setting Environment Variables” on page 147 for more information.

Requirement:

To use the **NET START** command on a system that accesses an Informix data source, you must run the informix regcopy utility,

which is found in the %INFORMIXDIR%\bin directory. The **db2start** command does not require you to run regcopy.

7. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps, such as adding server options and creating a user mapping are not required at this time; but, they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.

Sample DDL statements for an Informix data source are:

```
CREATE SERVER OPTION colseq FOR SERVER infrmxb
SETTING 'I'

CREATE USER MAPPING FROM marlow TO SERVER infrmxb
AUTHID "bigdog" PASSWORD "growler"
```

8. Create nicknames. When you create nicknames, remember to put double quotes around the Informix schema and table name if they are case sensitive. For example:

```
CREATE NICKNAME sanjose.employees FOR infmx."st1"."employees"
```

See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about the CREATE NICKNAME statement.

9. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in step 7 on page 84. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 11. Accessing Microsoft SQL Server Data Sources Using ODBC



A TaskGuide is available to help you easily configure Microsoft SQL Server data sources through ODBC, or you can perform this step manually using the following instructions.

To configure DataJoiner access to a Microsoft SQL Server data source:

1. If Microsoft SQL Server is not installed at the DataJoiner server workstation, install the Microsoft SQL Server ODBC client at the DataJoiner server workstation.
2. If you installed the Microsoft SQL Server client software, configure it using the Microsoft SQL Client Configuration utility.
3. Make sure that the SQL Server is registered as an ODBC system DSN:
 - a. Open the NT Control Panel (**My Computer** -> **Control Panel**)
 - b. Double-click on the ODBC icon
 - c. Click on **System DSN...**; the System Data Source window opens
 - d. Add or find (and then write-down) the name of the Microsoft SQL Server data source. When adding a data source name, ensure that it is a System DSN.

4. Use the CREATE SERVER MAPPING DDL statement to define the new Microsoft SQL Server data source. When performing server mapping:
 - Choose a unique server name.
 - Set NODE to the Microsoft SQL Server data source system DSN. This value is case sensitive.
Note that you do not need to use the **CATALOG NODE** command for data sources that are accessed through ODBC.
 - Set DATABASE to the name of a database at the server that is listed as the value for NODE. This value is case sensitive.
 - Set TYPE to mssqlserver.
 - Set VERSION to the data source version that the server is running. For example, 6.0.
 - Set PROTOCOL to "djmssql". This value is case sensitive.

An example statement is:

```
CREATE SERVER MAPPING FROM mssoft TO NODE "LocalServer"  
DATABASE "test_db" TYPE mssqlserver VERSION 6.0 PROTOCOL "djmssql"
```

Attention:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"test_db\".

5. If you want the Microsoft SQL Server data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM variable. For example, if you are using the data access module named djxmssql, the entry would be:

```
DJXCOMM=djxmssql
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

6. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps (adding server options, creating a user mapping, creating a nickname) are not required at this time; but they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.

Sample DDL statements for Microsoft SQL Server are:

```
CREATE SERVER OPTION colseq FOR SERVER mssoft  
SETTING 'I'
```

```
CREATE USER MAPPING FROM marlow TO SERVER mssoft  
AUTHID "bigdog" PASSWORD "growler"
```

```
CREATE NICKNAME dbo.sysobjects FOR mssoft.\"dbo\".\"sysobjects\"
```

7. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in the previous step. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.

- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 12. Accessing Oracle RDB Data Sources Using ODBC

To configure DataJoiner access to an Oracle RDB data source:

1. Make sure that the RDB server is registered as an ODBC system DSN:
 - a. Open the NT Control Panel (**My Computer** -> **Control Panel**).
 - b. Double-click on the ODBC icon.
 - c. Click on **System DSN...**; the System Data Source window opens.
 - d. Add or find (and then write-down) the name of the RDB server data source. When adding a data source name, ensure that it is a System DSN.
2. Use the CREATE SERVER MAPPING DDL statement to define the RDB server data source. When performing server mapping:
 - Choose a unique server name.
 - Set NODE to the RDB server data source system DSN. This value is case sensitive.

Note that you do not need to use the **CATALOG NODE** command for data sources that are accessed through ODBC.
 - Set DATABASE to the name of a database at the server that is listed as the NODE value. This value is case sensitive.
 - Set TYPE to rdb.
 - Set VERSION to the data source version that the server is running. For example, 6.0.
 - Set PROTOCOL to "djxrd". This value is case sensitive.

An example statement is:

```
CREATE SERVER MAPPING FROM rdb1 TO NODE "RDBServer"  
DATABASE "test_db" TYPE rdb VERSION 6.0 PROTOCOL "djxrd"
```

Attention:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
 - From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"test_db\".
3. If you want the Oracle RDB data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of

the DataJoiner instance to include an entry for the DJXCOMM variable. For example, if you are using the data access module named djxrdb, the entry would be:

```
DJXCOMM=djxrdb
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

4. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps (adding server options, creating a user mapping, creating a nickname) are not required at this time; but they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.

Sample DDL statements for RDB are:

```
CREATE SERVER OPTION colseq FOR SERVER rdb1  
SETTING 'I'
```

```
CREATE USER MAPPING FROM marlow TO SERVER rdb1  
AUTHID "bigdog" PASSWORD "growler"
```

```
CREATE NICKNAME dbo.sysobjects FOR rdb1.\\"dbo\\".\\"sysobjects\\"
```

5. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in the previous step. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 13. Accessing Oracle Data Sources Using SQL*Net or Net8

Two different Oracle data access modules are included with DataJoiner: one for use with the SQL*Net V1 or V2 client software, and one for use with the Net8 client software. The client software you use dictates which data access module you will use. If you use SQL*Net, you must use the sqlnet data access module, and if you use Net8, you must use the net8 data access module. Regardless of the client software you use, however, you can access both Oracle Version 7 and Oracle Version 8 data sources.



A TaskGuide is available to help you easily accomplish this step, or you can perform this step manually using the following instructions.

To manually set up access to Oracle databases:

1. Install Oracle SQL*Net or Net8 on the DataJoiner system.
2. Configure SQL*Net or Net8. See the Oracle documentation for configuration instructions.

Complete Step 3 through Step 6 for each DataJoiner instance that will use the Oracle data access module.

3. If not already done, update the system environment variables with Oracle information:
 - Set the ORACLE_HOME environment variable to the Oracle home directory.
SQL*Net and Net8 require that this variable be set prior to starting your DataJoiner instance. If this variable is changed, the DataJoiner instance must be stopped and restarted for the new ORACLE_HOME value to take effect.
The value of ORACLE_HOME that you set for the DataJoiner instance is the one that is used by DataJoiner. If an individual user of DataJoiner has the ORACLE_HOME environment variable set, it is not used by DataJoiner.
4. Ensure that the SQL*Net or Net8 tnsnames.ora file (usually found in the %ORACLE_HOME%\NETWORK\ADMIN directory for SQL*Net and in the %ORACLE_HOME%\NET8\ADMIN directory for Net8) is updated for each Oracle server to which communications are configured. The SID is the name of the Oracle instance. The HOST is the host name where the Oracle server is located.

5. If the TNS_ADMIN environment variable is set in the environment of the DataJoiner instance, it will be used to locate the tnsnames.ora file. TNS_ADMIN is optional.
6. Recycle (**NET STOP *instance name*** followed by **NET START *instance name***) the DataJoiner instance.
7. Use the CREATE SERVER MAPPING DDL statement to define each Oracle server to which communications are configured. When performing server mapping:
 - Choose a unique server name.
 - Set NODE to the appropriate database server name from the tnsnames.ora file. This value is case sensitive. For an example, see Figure 10 on page 94.
Note that you do not need to use the **CATALOG NODE** command for Oracle data sources.
 - Set TYPE to oracle.
 - Set VERSION to the data source version that the server is running. For example, 7.2 for Oracle 7.2.
 - Set PROTOCOL to "sqlnet" if using Oracle's SQL*Net client software, or to "net8" if using Oracle's Net8 client software. This value is case sensitive.

A sample entry for an Oracle data source is:

```
CREATE SERVER MAPPING FROM oracle1a TO NODE "orala" TYPE oracle
VERSION 7.2 PROTOCOL "sqlnet"
```

A DATABASE value is not required; Oracle does not have multiple databases per instance to which you must connect to access data.

Preserving case-sensitivity:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
 - From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"sqlnet\".
8. If you want the Oracle data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM variable. For example, if you are using the data access module named sqlnet, the entry would be:

```
DJXCOMM=sqlnet
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

9. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps (adding server options, creating a user mapping, creating a nickname) are not required at this time; but, they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.

Sample DDL statements are:

```
CREATE SERVER OPTION colseq FOR SERVER oracle1a
SETTING 'N'
CREATE USER MAPPING FROM marlow TO SERVER oracle1a
AUTHID "bigdog" PASSWORD "growler"
CREATE NICKNAME dogpound FOR oracle1a.bigdog.dogpound
```

Attention:

The Oracle user (at the Oracle data source, not DataJoiner), must have been created using the Oracle **create user** command with the ‘identified by’ clause rather than the ‘identified externally’ clause.

Figure 10 on page 94 shows the information in the tnsnames.ora file and SYSCAT.SERVERS view. For Oracle, the value of dbname is a null string (“”) because Oracle does not have multiple databases in one instance of its database manager.

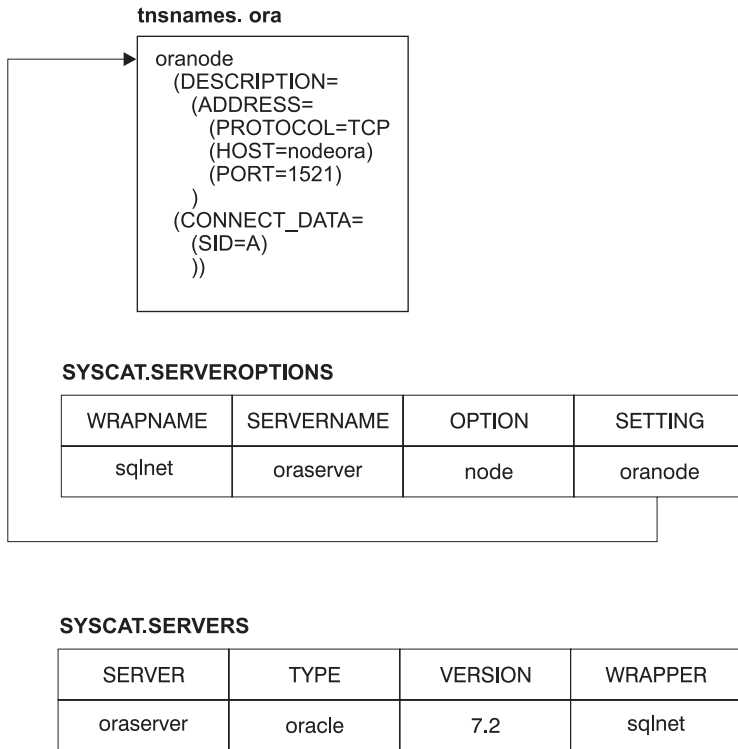


Figure 10. ORACLE Data Access Module

- For each 'HOST' in the DESCRIPTION section of the tnsnames.ora file, update the TCP/IP hosts file if necessary.

Whether you must update the TCP/IP hosts file depends on how TCP/IP is configured on your network. Part of the network must translate the remote hostname specified in the DESCRIPTION section in the tnsnames.ora file ("nodeora" in our example) to an address. If your network has a name server that recognizes the hostname, you do not need to update the TCP/IP hosts file. Otherwise, you need an entry for the remote host. See your network administrator to learn how your network is configured.

- Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in step 9 on page 93. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 14. Accessing Sybase Data Sources Using Sybase Open Client

The Sybase data access module is included with DataJoiner for Windows NT and supports the ctlib access mechanism.



A TaskGuide is available to help you easily configure Sybase data sources, or you can perform this step manually using the following instructions.

Follow these steps to manually configure the Sybase data access module:

1. Install catalog stored procedures on the Sybase node. In the Sybase documentation, see the *Command Reference Manual for Sybase SQL Server* for details.
2. Install the Sybase Open Client libraries on the DataJoiner system.

Complete steps 3 through 6 for each DataJoiner instance that will use the Sybase data access module.

3. If not already done, update the system environment variables with SYBASE information:
 - a. Set the SYBASE variable to the Sybase home directory.
 - b. Set the PATH variable so that it includes the Sybase BIN directory.
4. Recycle (**NET START *instance name* NET STOP *instance name***) the DataJoiner instance.
5. Use the Sybase-supplied utility to create an interfaces file that includes the data for all the Sybase Open Servers you want to access. This file is typically named `sql.ini`. See the *Sybase System Administration Guide* for more information on this procedure.

Figure 11 on page 98 contains an example of a Sybase Open Client interfaces file:

Sybase Open-Client interfaces file

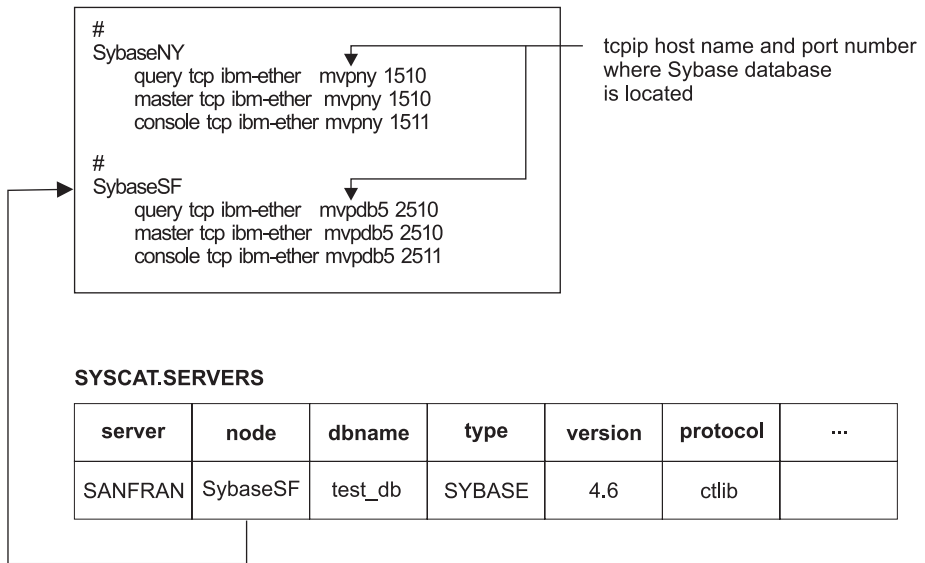


Figure 11. Sybase Data Access Module. The database server name in the Open Client interfaces file corresponds to the NODE column in SYSCAT.SERVERS; the host machine name and port number correspond to the TCP/IP host machine name and port address for the Sybase data source.

6. Rename the file you just created from sql.ini to interfaces. This step is not required, but if you choose not to rename sql.ini to interfaces, you must use the IFILE parameter of the CONNECTSTRING option that is explained in “Specifying a Connect String for Open Client” on page 100 to make the file available to Sybase.
7. Make the interfaces file accessible by doing *one* of the following tasks:
 - Put the file in the DataJoiner instance’s %DB2PATH% directory.
 - Specify the path and name of the interfaces file by using CONNECTSTRING in SYSCAT.SERVER_OPTIONS. For information about CONNECTSTRING, see “Specifying a Connect String for Open Client” on page 100.
8. Use the CREATE SERVER MAPPING DDL statement to define each database you want to access using Open Server Continue. When performing server mapping:
 - Choose a unique server name.
 - Set NODE to the appropriate database server name from the Open Client interfaces file. This value is case sensitive. For an example, see Figure 11.Note that you do not need to use the CATALOG NODE command for data sources that are accessed through Open Client.

- Set DATABASE to the name of a database at the server that is listed as the NODE value. This value is case sensitive.
- Set TYPE to sybase.
- Set VERSION to the data source version that the server is running. For example, 4.3 for Sybase 4.3.
- Set PROTOCOL to "ctlib". This value is case sensitive.

A sample entry for a Sybase data source is:

```
CREATE SERVER MAPPING FROM sybase1 TO NODE "sybase1" DATABASE "test_db"
TYPE sybase VERSION 4.3 PROTOCOL "dblib"
```

Notice that the server value is the same as the node value. The NODE value must match the server name in the Open Client interfaces file.

Preserving case-sensitivity:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"test_db\".

9. If you want the ctlib or dblib data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM variable. For example, if you are using both the ctlib and dblib data access modules, the entry would be:

```
DJXCOMM=ctlib,dblib
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

10. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps (adding server options, creating a user mapping, creating a nickname) are not required at this time; but, they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.

Sample DDL statements for a Sybase data source are:

```
CREATE SERVER OPTION colseq FOR SERVER sybase1
SETTING 'I'
CREATE USER MAPPING FROM marlow TO SERVER sybase1
AUTHID "bigdog" PASSWORD "growler"
```

```
CREATE NICKNAME dbo.sysobjects FOR mssoft.\\"dbo\".\\"sysobjects\"
```

11. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in the previous step. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Specifying a Connect String for Open Client

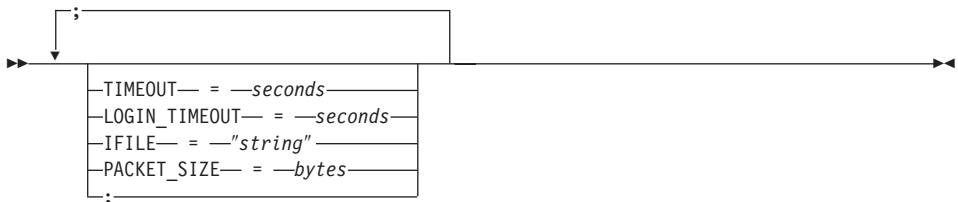
Open Client uses timeout thresholds to interrupt queries that run for too long a period of time. You can set these thresholds in DataJoiner by using the CONNECTSTRING option of the CREATE SERVER OPTION DDL statement.

Use the CONNECTSTRING option to specify:

- Timeout duration for SQL queries
- Timeout duration for login response
- Path and name of an interfaces file to use
- Packet size

Syntax

The syntax of CONNECTSTRING entries is:



TIMEOUT

Specifies the number of seconds for DataJoiner to wait for a response from Open Client for any SQL statement. The value of *seconds* is a positive whole number in DataJoiner's integer range or the default value, which for ctlib is -9999. This default value causes DataJoiner to wait indefinitely for a response.

LOGIN_TIMEOUT

Specifies the number of seconds for DataJoiner to wait for a response from Open Client to the login request.

IFILE

Specifies the path and name for the Open Client interfaces file to use. The path that is identified in *string* must be enclosed in double quotes ("). The default is %DB2PATH% in the home directory of your DataJoiner instance.

PACKET_SIZE

Specifies the packet size in bytes. If the data source does not support the specified packet size, the connection will fail. Increasing the packet-size when each record is very large (for example, when inserting rows into large tables) significantly increases performance. The byte size is a numeric value. See the Sybase reference manuals for more information.

These keywords are not case sensitive. The parameters can be specified in any order, but can be specified only once. The semicolon (;) must be present for each parameter and its value. The equal sign (=) must be present. There can be any number of tabs or spaces that separate the parameter from the equal sign, the equal sign from the value, and the value from the semicolon.

Examples

To set the timeout value to 60 seconds, set the connect string to:

```
TIMEOUT=60;
```

To set the timeout value to 60 seconds and the interfaces file to C:\etc\interfaces, set the connect string using the following example:

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1  
SETTING 'TIMEOUT=60;IFILE="C:\etc\interfaces";'
```

Chapter 15. Accessing Sybase SQL Anywhere Data Sources Using ODBC

To configure DataJoiner access to a Sybase SQL Anywhere data source:

With DataJoiner installed, complete the following steps:

1. Install your ODBC driver on the Windows NT system where DataJoiner is installed.
2. Use the Microsoft ODBC Data Source Administrator to add data sources for your ODBC driver. Make sure that you add them as System DSNs.
3. Perform any configuration that is required by SQL Anywhere. This task might include defining communication paths or database aliases.
4. Issue a CREATE SERVER MAPPING DDL statement for each data source you added in step 2.
 - Choose a unique server name.
 - Set NODE to the ODBC data source name. This value is case sensitive.
 - Set DATABASE to the name of a database at the server that is listed in the NODE column. This value is case sensitive.
 - For TYPE, specify sqlanywhere.
 - For VERSION, specify 5.0.
 - For PROTOCOL, specify sqlany. This field is case sensitive.

A sample entry for a SQL Anywhere data source looks like this:

```
CREATE SERVER MAPPING FROM bradgn1 TO NODE "odbcds1" DATABASE "test_db"  
TYPE sqlanywhere VERSION 5.0 PROTOCOL "sqlany"
```

Attention:

The syntax required to preserve case-sensitivity depends on where you are entering the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
 - From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"test_db\".
5. If you want the Sybase SQL Anywhere data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM variable. For example, if you are using the sqlany data access module, the entry would be:

```
DJXCOMM=sqlany
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

6. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps, such as adding server options and creating a user mapping, are not required at this time; but, they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* and the *DataJoiner Administration Supplement* for more information about DDL statements and when to use them.

Sample DDL statements are:

```
CREATE SERVER OPTION colseq FOR SERVER bradgn1  
SETTING 'I'
```

```
CREATE USER MAPPING FROM brad TO SERVER bradgn1  
AUTHID "bhauth1" PASSWORD "power2u"
```

7. Create nicknames. For example:

```
CREATE NICKNAME dept FOR bradgn1.bhauth1.dept
```

See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about the CREATE NICKNAME statement.

8. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping. See the *DataJoiner Application Programming and SQL Reference* for more information.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 16. Accessing Teradata Data Sources

DataJoiner for Windows NT supports Teradata Version 2.0.1 or later. Windows NT clients require Teradata CLIV2 (WinCLI) version 04.04.02 or higher. The client version can be displayed by using the Teradata BTEQ utility. Enter the BTEQ command **show version** to display the CLIV2 version.

Note:

- Due to Teradata restrictions, DataJoiner automatically issues a COMMIT WORK after every transparent DDL statement (for example, CREATE TABLE, ALTER TABLE, DROP TABLE), even when AUTOCOMMIT is OFF.

To access Teradata databases:

1. Set up the Teradata TCP/IP hosts file in your `\winnt\system32\drivers\etc` directory. You need to set up this file on each client before you can successfully request a connection to a remote Teradata server. For example, the hosts file should contain a line similar to this:

```
192.168.100.100 tdatsvr.companyname.com tdatsvrCOP1 # Remote Teradata Server
```

where:

```
192.168.100.100
```

Is the TCP/IP address of the remote server.

```
tdatsvr.companyname.com
```

Is a symbolic name associated with the remote server.

```
tdatsvrCOP1
```

Is the alias for the remote server. The alias must begin with an alphabetic string and end with the `COP n` suffix, where n is a number between 1 and the total number of applications processors (APs) associated with the Teradata communications processor (COP).

Figure 12 on page 106 shows the general structure of a hosts file.

Teradata hosts file

TCP/IP address	Host name	Alias
192.168.100.100	tdatsvr.companyname.com	tdatsvrCOP1

SYSCAT.SERVERS view

server	node	type	version	protocol
TDAT	tdatsvr	TERADATA	2.0.1	teradata

Figure 12. Teradata Data Access Module

See the *Teradata WinCLI Installation Guide* for more information about the hosts file.

2. Install the Teradata client library (WINCLI32.DLL) on the system where DataJoiner is installed.

Complete steps 3 through 7 for each DataJoiner instance that will use the Teradata data access module.

3. Update the ENVIRONMENT of the DataJoiner instance to include the Teradata COPLIB and COPERR environment variables; for example:

```
COPLIB=teradata-lib-directory  
COPERR=teradata-lib-directory
```

where *teradata-lib-directory* is the directory in which you installed the Teradata client library in step 2. See the *Teradata WinCLI Installation Guide* for more information on the COPLIB and COPERR environment variables.

4. Use the CREATE SERVER MAPPING DDL statement to define each Teradata server that you expect to access via DataJoiner. For the CREATE SERVER MAPPING statement:

- Choose a unique server name.
- Set NODE to the appropriate server alias from the hosts file (excluding the suffix COPn). For an example, see Figure 12.

You do not need to use the DataJoiner **CATALOG NODE** command for any data sources that are accessed through the Teradata data access module.

- Set TYPE to teradata.
- Set VERSION to the version of Teradata that the Teradata server is running. For example, use 2.0.1 for Teradata 2.0.1 (only one-digit numbers and two-digit numbers separated by a decimal point are supported).
- Set PROTOCOL to "teradata". (See Figure 12 on page 106).

A sample CREATE SERVER MAPPING statement for Teradata is:

```
CREATE SERVER MAPPING FROM tdat TO NODE "tdatsvr"
TYPE teradata VERSION 2.0.1 PROTOCOL "teradata"
```

Preserving case-sensitivity:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.
 - From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"teradata\".
5. Use additional DataJoiner DDL statements to refine access to the data sources that you defined with the CREATE SERVER MAPPING statement. Other steps (adding server options, creating user mappings, creating nicknames) are not required at this time, but they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about DDL statements and when to use them.
 6. If you want the Teradata data access module to be loaded automatically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM environment variable.

The name of the Teradata data access module that you created should be specified in the DJXCOMM variable. For example, if you are using the data access module that is named teradata, the entry would be:

```
DJXCOMM=teradata
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

7. Recycle (**db2stop** followed by **db2start**) the DataJoiner instance.
8. Optional: Test the connections to your data sources. This step is recommended because it allows you to determine if your data source is configured correctly.

- a. If the user ID and password used to access the Teradata data source are different from the user ID and password used at the DataJoiner instance, you must create a user mapping as mentioned in step 5 on page 107, for example:

```
CREATE USER MAPPING FROM USER TO SERVER TDAT AUTHID WILSON
PASSWORD SECRET
```

See the *DataJoiner Application Programming and SQL Reference* for more information.

- b. Connect to the DataJoiner database that you created in “Creating a DataJoiner Database” on page 43.
- c. Either create a nickname for a table at a Teradata server, or use pass-through to directly access a Teradata table.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Default Type Mappings from Teradata Data Sources to DataJoiner

This section shows default forward mappings between DB2 for CS data types defined to DataJoiner and Teradata data types. In these mappings, the REMOTE_TYPENAMES defined in the Teradata database are mapped to the corresponding TYPENAMES in the DataJoiner database.

Table 6. Teradata Default Data Type Mappings in SYSCAT.SERVER_DATATYPES (Not All Columns Shown)

REMOTE_TYPENAME	REMOTE_META_TYPE	REMOTE_BIT_DATA	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_S_OPR_P	TYPENAME	BIT_DATA	LENGTH	SCALE
BYTEINT	S	-	-	-	-	-	-	SMALLINT	Y	2	-
SMALLINT	S	N	-	-	-	-	-	SMALLINT	Y	2	-
INTEGER	S	N	-	-	-	-	-	INTEGER	Y	4	-
DECIMAL	S	N	-	-	-	-	-	DECIMAL	Y	-	-
FLOAT	S	N	-	-	-	-	-	DOUBLE	Y	8	-

Table 6. Teradata Default Data Type Mappings in SYSCAT.SERVER_DATATYPES (Not All Columns Shown) (continued)

REMOTE_TYPENAME	REMOTE_META_TYPE	REMOTE_BIT_DATA	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_S_OPR_P TYPE	REMOTE_TYPENAME	BIT_DATA	LENGTH SCALE
CHAR	S	N	1	254	-	-	-	CHARACTER	N	-
CHAR	S	N	255	4000	-	-	-	VARCHAR	N	-
CHAR	S	N	4001	32000	-	-	-	LONG VARCHAR	N	32700
VARCHAR	S	N	1	4000	-	-	-	VARCHAR	N	-
VARCHAR	S	N	4001	32000	-	-	-	LONG VARCHAR	N	32700
BYTE	S	N	1	254	-	-	-	CHARACTER	Y	-
BYTE	S	N	255	4000	-	-	-	VARCHAR	Y	-
BYTE	S	N	4001	32000	-	-	-	LONG VARCHAR	Y	32700
VARBYTE	S	N	1	32000	-	-	-	VARCHAR	Y	-
VARBYTE	S	N	4001	32000	-	-	-	LONG VARCHAR	Y	32700
DATE	S	N	-	-	-	-	-	DATE	Y	4
GRAPHIC	S	N	1	127	-	-	-	GRAPHIC	N	-
GRAPHIC	S	N	128	2000	-	-	-	VARGRAPHIC	Y	-
GRAPHIC	S	N	2001	16000	-	-	-	LONG VARGRAPHIC	Y	16350
VARGRAPHIC	S	N	1	2000	-	-	-	VARGRAPHIC	Y	-
VARGRAPHIC	S	N	2001	16000	-	-	-	LONG VARGRAPHIC	Y	16350

Default Type Mappings from DataJoiner to Teradata Data Sources

This section shows default reverse mappings between DB2 for CS data types defined to DataJoiner and Teradata data types. In these mappings, the TYPENAMES defined to DataJoiner create the corresponding REMOTE_TYPENAMES defined in the Teradata database.

Table 7. Teradata Default Data Type Mappings in SYSCAT.REVTYPENAME MAPPINGS (Not All Columns Shown)

REMOTE_TYPENAME	REMOTE_META_TYPE	REMOTE_BIT_DATA	REMOTE_LENGTH	REMOTE_SCALE	TYPENAME	BIT_DATA	LOCAL_LOWER_LEN	LOCAL_UPPER_LEN	LOCAL_LOWER_SCALE	LOCAL_UPPER_SCALE	LOCAL_S_OPR_P
SMALLINT	S	N	-	-	SMALLINT	Y	-	2	-	-	-
INTEGER	S	N	-	-	INTEGER	Y	-	4	-	-	-
DECIMAL	S	N	-	-	DECIMAL	Y	1	18	0	18	-
FLOAT	S	N	-	-	DECIMAL	Y	19	31	0	31	-
FLOAT	S	N	-	-	DOUBLE	Y	-	8	-	-	-
CHAR	S	N	-	-	CHARACTER	N	-	-	-	-	-
VARCHAR	S	N	-	-	VARCHAR	N	-	-	-	-	-
LONG VARCHAR	S	N	-	-	LONG VARCHAR	N	-	32700	-	-	-
BYTE	S	N	-	-	CHARACTER	Y	-	-	-	-	-
VARBYTE	S	N	-	-	VARCHAR	Y	-	-	-	-	-
VARBYTE	S	N	32000	-	LONG VARCHAR	Y	-	32700	-	-	-
DATE	S	N	-	-	DATE	Y	-	4	-	-	-
FLOAT	S	N	-	-	TIME	Y	-	3	-	-	-
DECIMAL	S	N	18	4	TIMESTAMP	Y	-	10	-	-	-
GRAPHIC	S	N	-	-	GRAPHIC	N	-	-	-	-	-
VARGRAPHIC	S	N	-	-	VARGRAPHIC	Y	-	-	-	-	-
LONG VARGRAPHIC	S	N	-	-	LONG VARGRAPHIC	Y	-	16350	-	-	-

Chapter 17. Accessing IMS and VSAM Data Sources Using Classic Connect

Classic Connect is a separately-orderable component of DataJoiner that provides read access to nonrelational data stored in Information Management Systems (IMS) databases and Virtual Storage Access Method (VSAM) data sets.

For information about configuring DataJoiner and Classic Connect to access IMS and VSAM data, see the *DB2 DataJoiner Classic Connect Installation, Configuration, and Reference Guide V2.1.1*.

Chapter 18. Accessing Data Sources Using a Generic Data Access Module

Each data source that uses the generic access API appears to be a DBMS to DataJoiner, whether it is actually a DBMS or not. See the *DataJoiner Generic Access API Reference* for information on generic data access modules.

Restriction:

The generic access API supports ODBC and X/Open CLI compliant drivers. If the driver you are using does not meet ODBC or X/Open CLI specifications for API calls, SQL dialect, and SQLSTATES, the generic access data access module might not function properly.

With DataJoiner installed, complete the following steps:

1. Install your ODBC driver on the Windows NT system where DataJoiner is installed.
2. Use the Microsoft ODBC Data Source Administrator to add data sources for your ODBC driver. Make sure that you add them as System DSNs.
3. Perform any configuration that is required by the data access module provider. This task might include defining communication paths or database aliases. See the documentation provided by the data access module provider.
4. Issue a CREATE SERVER MAPPING DDL statement for each data source you added in step 2.
 - Choose a unique server name.
 - Set NODE to the ODBC data source name. This value is case sensitive.
 - Set DATABASE to the name of a database at the server that is listed in the NODE column. This value is case sensitive.
 - For TYPE, specify generic.
 - For VERSION, specify 2.1.
 - For PROTOCOL, specify generic. This value is case sensitive.

A sample entry for a generic data source looks like this:

```
CREATE SERVER MAPPING FROM bradgn1 TO NODE "odbcds1" DATABASE "test_db"  
TYPE generic VERSION 2.1 PROTOCOL "generic"
```

Preserving case-sensitivity:

The syntax required to preserve case-sensitivity depends on where you enter the CREATE SERVER MAPPING statement.

- From the DB2 interactive mode command prompt, you need to enclose all case-sensitive parameters in double quotes.

- From a Windows NT operating system command prompt, you need to enclose all case-sensitive parameters in the backslash, double quote (\") character sequence; for example, \"test_db\".
5. If you want the generic data access module to start dynamically when the DataJoiner instance is started, update the Windows NT environment of the DataJoiner instance to include an entry for the DJXCOMM variable. For example:

```
DJXCOMM=generic
```

Ensure that there are no spaces on either side of the equal sign (=).

See “Chapter 21. Setting Environment Variables” on page 147 for more information.

6. Use additional DataJoiner DDL statements to refine access to the data sources you defined using CREATE SERVER MAPPING statements. Other steps, such as adding server options and creating a user mapping, are not required at this time, but they are suggested. See the *DataJoiner Application Programming and SQL Reference Supplement* and the *DataJoiner Administration Supplement* for more information about DDL statements and when to use them.

Sample DDL statements are:

```
CREATE SERVER OPTION colseq FOR SERVER bradgn1
SETTING 'I'

CREATE USER MAPPING FROM brad TO SERVER bradgn1
AUTHID "bhauth1" PASSWORD "power2u"
```

7. Create nicknames. For example:

```
CREATE NICKNAME dept FOR bradgn1.bhauth1.dept
```

See the *DataJoiner Application Programming and SQL Reference Supplement* for more information about the CREATE NICKNAME statement.

8. Test the connections to your data sources. This step is optional but recommended because it allows you to determine if your data source is configured correctly.

Prerequisite step:

Before testing, you must perform user mapping as mentioned in step 6.

To test your location transparency connections through DataJoiner, complete the following tasks:

- a. Connect to the DataJoiner database.
- b. Either create a nickname with DataJoiner for a table at a data source, or use pass-through to directly access a table at a data source.

Refer to the *DataJoiner Application Programming and SQL Reference Supplement* for more information about creating nicknames and using pass-through.

Chapter 19. Configuring Communications for DRDA Data Sources

This chapter describes how you configure communications between a DataJoiner workstation and DRDA servers by using the DDCS functionality that is provided in DataJoiner.

The DDCS function that is provided in DataJoiner is equivalent to DDCS 2.4. You do not need to install DDCS if you have installed DataJoiner.

You cannot have two versions of a DB2 product installed on the same NT workstation; therefore, you cannot have DDCS and DataJoiner installed on the same NT workstation.

DRDA data sources can be accessed via SNA LU 6.2, and DRDA3 data sources can be accessed through SNA LU 6.2 or TCP/IP.

Before you perform any of the tasks that are listed in this chapter, you must:

- Have installed DataJoiner, TCP/IP, IBM Communications Server, or Microsoft SNA Server, and an appropriate Link Service. For more information, see “Software Requirements” on page 25.
- Have access to a DRDA server host that has DRDA Application Server configured and operational; for DB2/MVS this is DDF.

For details of how to configure a DB2 for MVS host, refer to the *DDCS Installation and Configuration Guide*. For details on how to configure an AS/400, VSE, or VM host, or for more information about DDCS, refer to the *Distributed Relational Database Architecture Connectivity Guide*.

- Have access to a VTAM administrator (if you are configuring DataJoiner DDCS via SNA) and a host database administrator for MVS, VSE, or VM databases, or have access to an AS/400 administrator for OS/400 databases.
- Use a token ring network as the communications medium.

Configuring DataJoiner as a DRDA Application Server

Instructions for configuring DataJoiner as a DRDA Application Server are included in the following steps. Some steps are specific to an AS configuration; some steps are not needed for an AS configuration; and some steps are different for an AS configuration. These differences are noted where applicable.

Authentication Restriction:

If you choose APPC as your communication protocol, the communications subsystem might restrict the types of database manager configuration authentication settings you can use at the DataJoiner server. Not all communications subsystems expose the application requester's password to the application server when security (program) is used. When this is the case, the database manager configuration authentication must not be set to SERVER. This restriction applies to all Windows NT communications subsystems.

To overcome this restriction that prevents you from using authentication SERVER, set the database manager authentication to DCS. This allows DRDA APPC connections that have been authenticated by the communication subsystem to succeed. However, with this setting the DataJoiner server will act as if authentication SERVER is being used for all other types of connections.

Configuring DataJoiner DDCS Using SNA

This section shows you how to access DRDA data sources via SNA LU 6.2 and contains the following steps:

- “Step 1: Gathering Information About the DRDA Servers”
- “Step 2: Gathering DDCS Information” on page 119
- “Step 3: Gathering Information for Each Server Connection” on page 120
- “Step 4: Configuring SNA LU 6.2” on page 120

Step 1: Gathering Information About the DRDA Servers

Before you can access a DRDA data source, you must configure both the DRDA server and the DataJoiner workstation. If you decide to configure the DataJoiner workstation before you configure the server, you must at least get information about how the server *will be* configured before you configure the DataJoiner workstation.

Use the following worksheets to gather information about the DRDA servers:

- “MVS, VSE, and VM Server Worksheet” on page 261
- “OS/400 Server Worksheet” on page 263.

For each server that you are connecting to, fill in one copy of the appropriate server worksheet.

For an MVS, VSE, or VM connection, you need to know the control point name of your machine. Your network administrator can provide you with this information.

Figure 13 illustrates the correspondences between information that is specified on the DRDA server system, information specified in the SNA Server Administration, and information specified in the directories. The values that are shown in this figure are the examples values that are used in this chapter. The instructions in this book show you how to update these values.

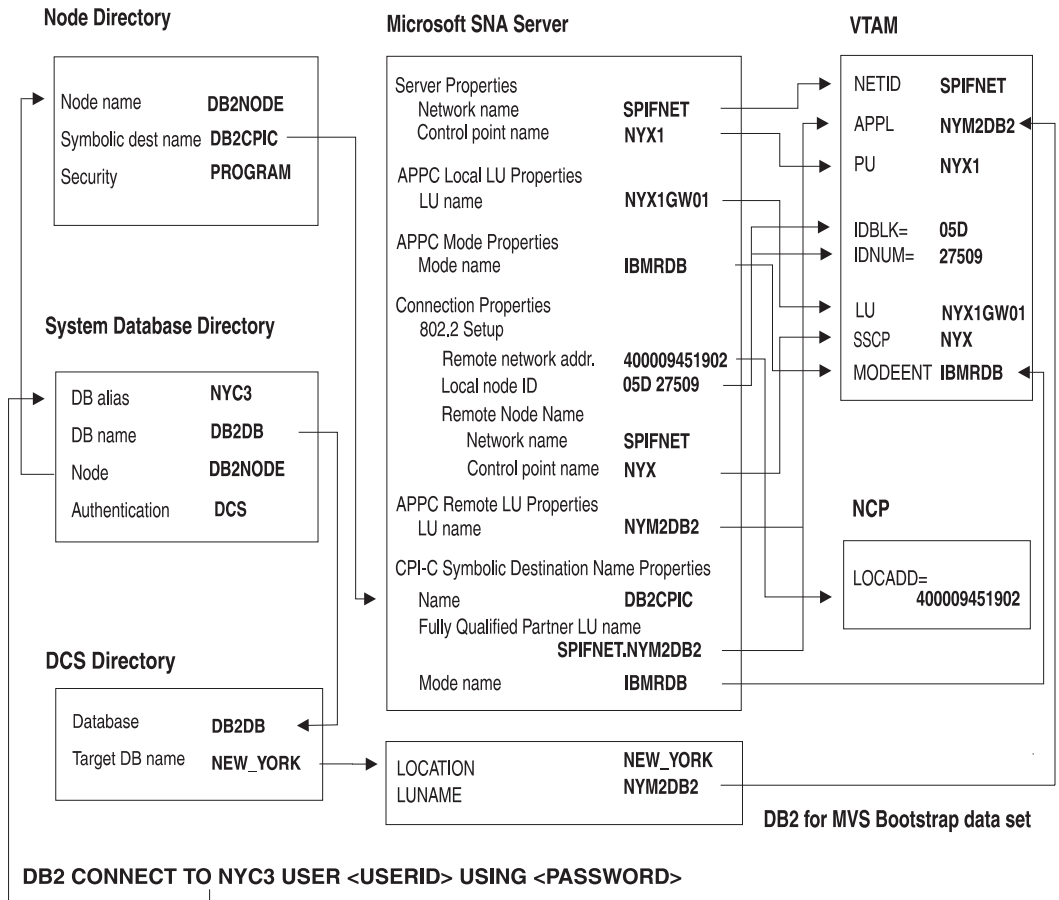


Figure 13. Configuration on the DRDA Server and DataJoiner Workstation

Step 2: Gathering DDCS Information

Fill in a copy of the “Windows NT Configuration Worksheet” on page 264.

For an MVS, VSE, or VM connection, perform the following steps:

1. For Network ID, copy the NETID (item 1) from the server worksheet.
2. For Local control point name, copy the PU (item 7) from the server worksheet.
3. For Local node ID, copy the IDBLK and IDNUM (item 8) from the server worksheet.
4. For Mode name, copy the MODEENT (item 3) from the server worksheet.
5. For Local LU name, copy the LU (item 6) from the server worksheet.
6. For Local LU alias, choose an alias. This can be the same as the LU name or different.

For an OS/400 connection, perform the following steps:

1. For Network ID, copy the local network ID (item 1) from the server worksheet.
2. For Mode name, copy the mode name (item 3) from the server worksheet.
3. Leave the other items blank for now.

Step 3: Gathering Information for Each Server Connection

For each server that you are connecting to, fill in a copy of the “Windows NT Connection to Server Worksheet” on page 264 as follows:

- For Link name, choose a value.
- For LAN destination address, copy item 2 (controller address or local adapter address) from the server worksheet.
- For Symbolic destination name, choose a value.
- For Partner CP name, copy item 4 from the server worksheet. For MVS, VSE, or VM it is the SSCP name; for OS/400 it is a local control point name.
- For Partner LU name for MVS, VSE, or VM, copy item 5 (APPL) from the server worksheet. For OS/400, copy the Partner CP name that you just wrote on this worksheet.
- For Remote transaction program, copy the value from the server worksheet. Use the value X'07F6C4C2' for MVS or OS/400, and use the RDB_NAME for VSE and VM.
- For Target database name, copy the last item from the server worksheet. This is a LOCATION value (MVS), RDB_NAME (VSE or VM), or relational database name (OS/400).

Step 4: Configuring SNA LU 6.2

You can configure your SNA LU 6.2 environment for access to DRDA data sources by using either of the following tools:

- IBM Communications Server; see “Configuring APPC for DataJoiner Using Communications Server” on page 121
- Microsoft SNA Server Manager; see “Configuring APPC for DataJoiner Using Microsoft SNA Server Manager” on page 124

The bold numbers that follow example values correspond to values you entered on the worksheets in “Appendix C. Worksheets for Use with DataJoiner DDCS” on page 261. For example, the number **3** is used to represent the Local Node ID from “Windows NT Configuration Worksheet” on page 264.

Configuring APPC for DataJoiner Using Communications Server

To use the APPC protocol, you need to create or customize SNA properties and server connection properties on the DataJoiner workstation. This section shows you how to perform these tasks by using IBM Communications Server Version 5.0.

Step 1: Configuring a Node: To begin configuring APPC communications using Communications Server, you need to open the Communications Server Node Configuration window:

1. Open the SNA Node Operations window (**Start**—> **Programs** —> **Communications Server** —> **Node Configuration**).
2. From the File menu, select **New**
3. From the Scenarios menu, select **Select CPI-C, APPC, or 5250 Emulation** to open the Communications Server Node Configuration window.
This is the main configuration window.
4. From the Communications Server SNA Node Configuration window, select **Configure Node**, then select **New**.
5. Enter the following required information. Use values from “MVS, VSE, and VM Server Worksheet” on page 261.

Fully qualified CP Name	SPIFNET.NYXGW01 1.6
CP alias	NYXGW01 6
Local Node ID	05D 27509 8
Node Type	Specify end node (default).

6. Click **OK** to continue.

Step 2: Configuring Devices:

1. From the SNA Node Configuration Window, click **Configure Devices**.
2. Click **New**.
3. Click **OK** to continue.

Step 3: Configuring Connections:

1. From the SNA Node Configuration Window, click **Configure Connection**, then click **New**.

2. On the Basic properties page, specify the following required value. Use the value you specified on “MVS, VSE, and VM Server Worksheet” on page 261.

Destination address 400009451902 2

3. On the Security properties page, specify the following required information. Use values you specified on “MVS, VSE, and VM Server Worksheet” on page 261.

Adjacent CP Name SPIFNET.NYX1 1.7

4. Click **OK** to create the connection definition.

Step 4: Configuring a Partner LU 6.2:

1. From the SNA Node Configuration Window, click **Configure Partner LU 6.2**, then click **New**.
2. On the Basic properties page specify the following required information. Use values you specified on the “MVS, VSE, and VM Server Worksheet” on page 261 and the “Windows NT Connection to Server Worksheet” on page 264.

Partner LU Name SPIFNET.NYM2DB2 1.11

Fully qualified CP Name SPIFNET.NYX 1.4

DRDA AS requirement:

If you are configuring DataJoiner as a DRDA AS, you also need to specify the Partner LU name and Fully qualified CP name of the application requester that will be calling the application server.

3. Click **OK** to continue.

Step 5: Configuring the Mode:

1. From the SNA Node Configuration Window, click **Configure Modes**, then click **New**.
2. In the Mode Name field, specify IBMRDB.
3. Click **OK** to continue.

Step 6: Configuring CPI-C Side Information:

1. From the SNA Node Configuration Window, click **Configure CPI-C Side Information**, then click **New**.
2. On the Basic property page, specify the following required information. Use values you specified on “MVS, VSE, and VM Server Worksheet” on page 261.

Symbolic destination name NYM2DB2 5

Mode name IBMRDB 3

Partner LU Name SPIFNET.NYM2DB2 1.5

TP name Default **9**

3. Click **OK** to continue.

Step 7: Configuring a Transaction Program Name:

DRDA AS consideration:

This step is required only if you are configuring DataJoiner as a DRDA AS.

1. From the SNA Node Configuration Window, click **Configure Transaction Programs**.
2. Click **New**.
3. Assign the TP name that is defined as the Transaction Program Name in the database manager configuration file.
4. Click the **Advanced** tab.
5. Deselect the following three check boxes:
 - PIP allow
 - Dynamically loaded
 - Full duplex support
6. Click **OK**.

Step 8: Saving and Testing the Configuration: Save your Communications Server configuration by selecting Exit from the File menu. The Save dialog window opens. Assign a file name with an extension of ".acg."

To test the configuration:

1. Click **Start Node** on the SNA Node Operations toolbar.
2. Select the configuration file you have just configured, and click **Open**.
3. Select **Connections** from the pull down menu on the toolbar.
4. If the state of the connection is Pending Active, select **Connections** again to refresh the display. Repeat this step until the connection becomes active..

When the connection is established, you should see a view of the connection that displays the control point name that you specified in Step 3.

5. To verify the connection, you can establish an LU 6.2 session to your destination:
 - a. From the toolbar, select **Operations**.
 - b. Select **CNOS Initialize**.
 - c. Enter the Partner LU 6.2 value you specified in Step 3.
 - d. Enter the Mode you specified in Step 4 (IBMRDB).
 - e. Click **OK**.
 - f. From the pull down menu, select **LU 6.2 Sessions**

You should see two LU 6.2 session entries.

If you need to change configuration values, select **Launch** —> **SNA Node Configuration**.

Configuring APPC for DataJoiner Using Microsoft SNA Server Manager

To use the APPC protocol, you need to create or customize SNA properties and server connection properties on the DataJoiner workstation.

The following sections provide a working example of the profiles that are configured on a DataJoiner workstation using Microsoft SNA Server Version 3.0, and show how to define SNA and server connection properties by using the **SNA Server Manager** tool.

Before you perform any of the configuration steps, make sure that you have completed the “Windows NT Configuration Worksheet” on page 264.

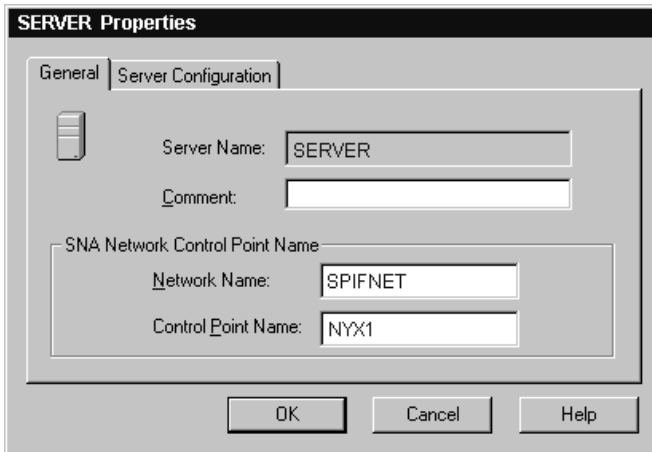
The boxed numerics correspond with the items that are located on the worksheet. For example, the symbol **3** would be used to represent the Local Node ID from “Windows NT Configuration Worksheet” on page 264. The ***** symbol is used to denote entries that need to be changed but do not have a representation on the worksheet.

Step 1: Installing the DLC Communications Protocol: Before you can begin the actual configuration process, you need to install the DLC communications protocol that is supplied on the Microsoft Developer Network CD. You should have received this CD with the Server version of Windows NT.

1. Select **Control Panel** —> **Protocols** —> **Add**. Click **DLC Protocol**.
2. Insert the Microsoft Developer Network CD into the CD-ROM drive.
3. Click **Install**.
4. When installation completes, shutdown and restart the computer.

Step 2: Defining Server Properties: Begin configuring the DataJoiner workstation by defining server properties:

1. Double-click the **Microsoft SNA Server** icon in the Windows NT Program Manager.
2. Double-click the **SNA Server Manager** icon. The SNA Server Manager window opens.
3. Double-click the machine name you are configuring. In this example, the machine name is SERVER. The Properties window opens for the specified server. By default, the **General** tab is displayed.



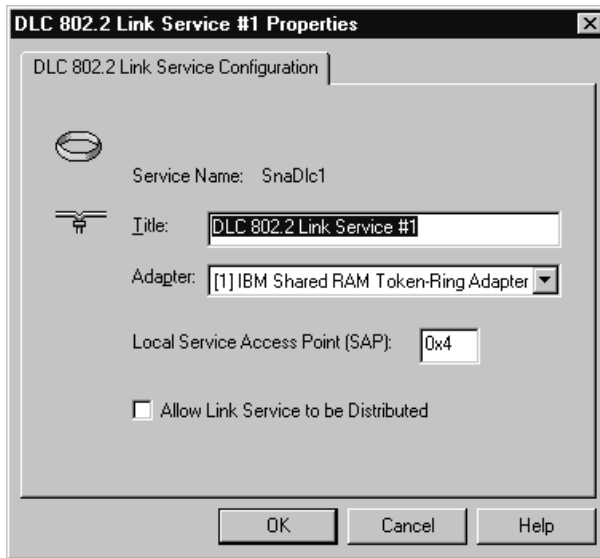
4. Enter the values you entered on “Windows NT Configuration Worksheet” on page 264 as follows:

Network Name	SPIFNET 1
Control Point Name	NYX1 2

5. Click **OK** to continue.

Step 3: Installing the Link Service:

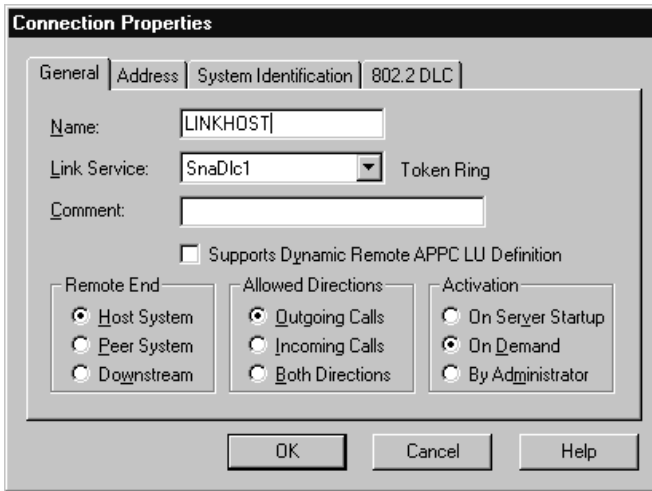
1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **Link Service**. The Insert Link Service window opens.
3. Select **DLC 802.2** and click **Add**. The DLC 802.2 Link Service Properties window opens:



4. Click **OK** to accept the default values.
5. Click **Finish**.

Step 4: Adding a Connection: To configure settings for a new connection, perform the following steps:

1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **Connection**.
3. Under **Connection**, select the type of connection. Our example uses 802.2. See the online help for information about the other types of connections. The Connection Properties window opens.



4. Use the following values (refer to “Windows NT Connection to Server Worksheet” on page 264):

For a connection to MVS, VM, or VSE, use the following values:

Name	LINKHOST 7
Link Service	SnaDlc1
Remote End	Host system radio button
Allowed Directions	Outgoing Calls radio button
Activation	On Demand radio button

For a connection to OS/400, use the following values:

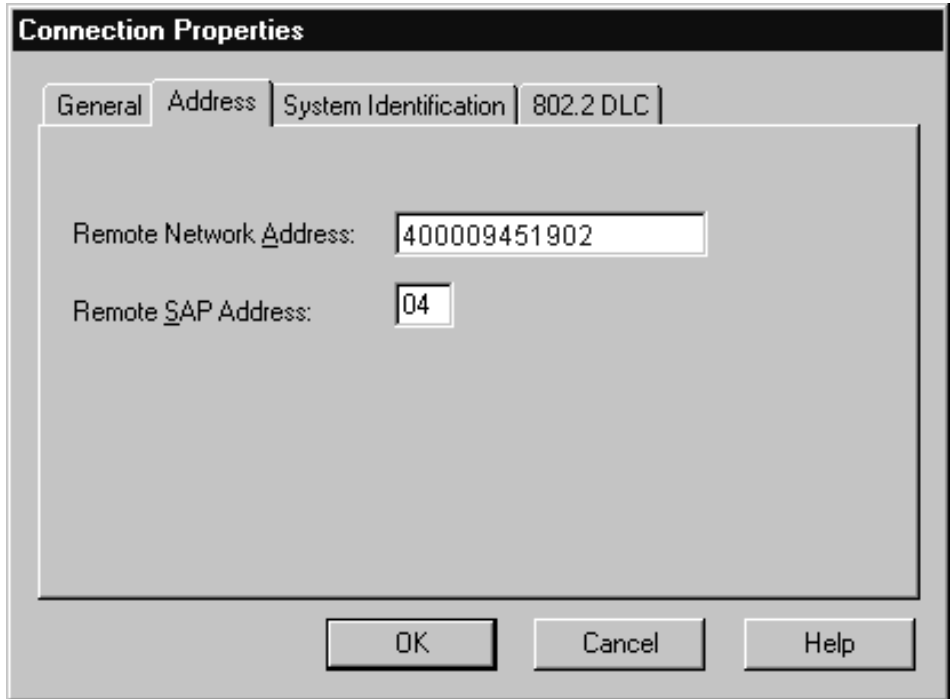
Name	LINKHOST 7
Link Service	SnaDlc1
Remote End	Peer System radio button
Allowed Directions	Outgoing Calls radio button
Activation	On Demand radio button

DRDA AS requirement:

If you are configuring DataJoiner as a DRDA AS, substitute the following settings for Remote End and Allowed Directions:

Remote End	Host System radio button
Allowed Directions	Both Directions radio button

5. Click the **Address** tab to display the window that is shown in the following figure:



For a connection to MVS, VM or VSE, use the values from “MVS, VSE, and VM Server Worksheet” on page 261, for example:

Remote Network Address 400009451902 2

Remote SAP Address 05D 27509 8

6. For a connection to OS/400, use the values from “OS/400 Server Worksheet” on page 263, for example:

Remote Network Address 400009451902 2

Remote SAP Address 04

7. Click the **System Identification** tab to display the window that is shown in the following figure:

Connection Properties

General | Address | System Identification | **802.2 DLC**

Local Node Name

Network Name: SPIFNET

Control Point Name: NYX1

Local Node ID: 05D 27509

XID Type

Format 0

Format 3

Remote Node Name

Network Name: []

Control Point Name: []

Remote Node ID: [] []

Peer DLC Role

Primary

Secondary

Negotiable

OK Cancel Help

For a connection to MVS, VM or VSE, use the values from “MVS, VSE, and VM Server Worksheet” on page 261, for example:

Network Name SPIFNET 1

Control Point Name NYX 4

Local Node ID 05D 27509 8

For a connection to OS/400, use the values from “OS/400 Server Worksheet” on page 263, for example:

Remote Network Address 400009451902 2

Control Point Name SYD2101A 4

Local Node ID 05D 27509

This value is arbitrary. If you also define a connection to MVS, VM, or VSE, you can use the same value as item 8 from the server worksheet.

DRDA AS requirement:

If you are configuring DataJoiner as a DRDA AS, you need to specify the following settings for the Remote Node Name. Use the

values from “Windows NT Configuration Worksheet” on page 264, and “Windows NT Connection to Server Worksheet” on page 264.

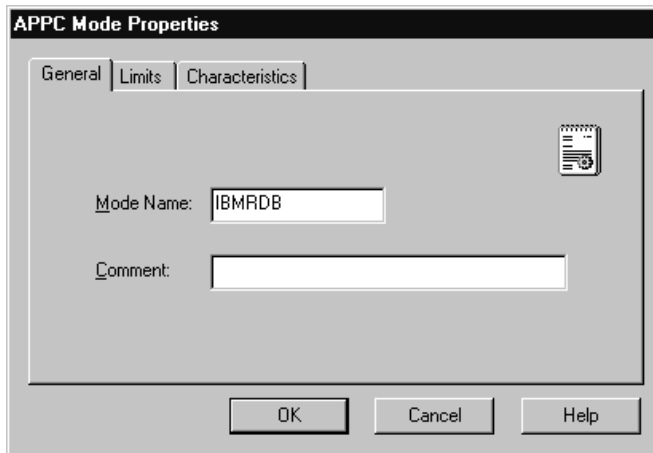
Network Name SPIFNET 1

Control Point Name NYX 10

8. Click **OK** to close the Connection Properties window.

Step 5: Set APPC Mode Properties: To configure a mode for the APPC LU, complete the following steps:

1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **APPC**.
3. Select **Mode Definition** to display the APPC Mode Properties window that is shown in the following figure. By default the **General** tab is displayed.

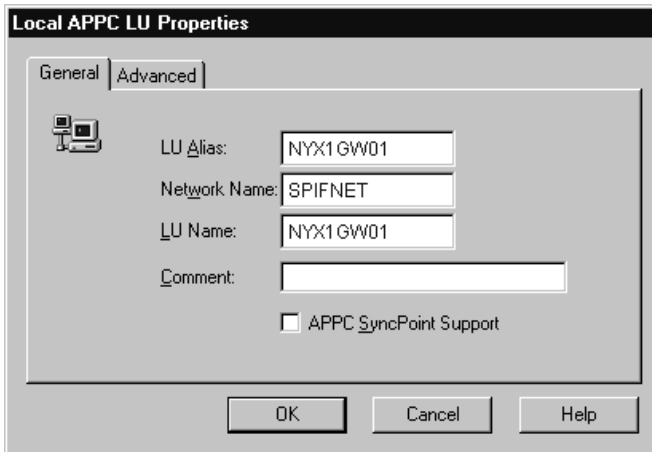


4. Enter **IBMRDB** in the **Mode Name** field.

The values in the Limits and Characteristics windows are arbitrary. The values for items such as **Parallel Session Limit** and **Minimum Contention Winner Limit** are for tuning purposes. See the online help for SNA Server for guidance on setting these values.

Step 6: Setting Local APPC LU Properties: To configure a mode for the Local APPC LU, complete the following steps:

1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **APPC**.
3. Select **Local LU** to display the Local APPC LU Properties window that is shown in the following figure. By default the **General** tab is displayed.



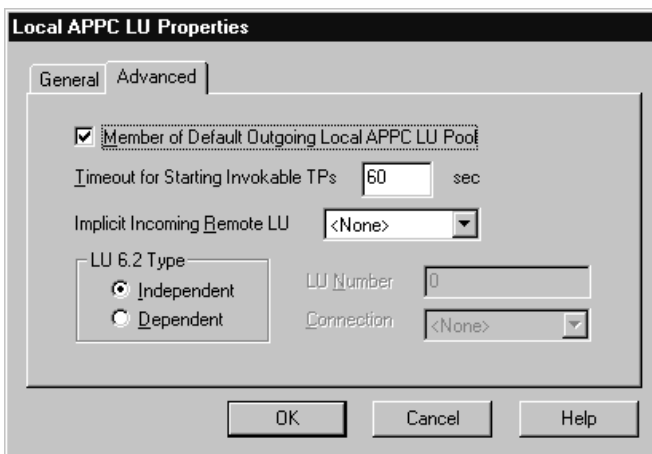
4. Enter the values you entered on “Windows NT Configuration Worksheet” on page 264:

LU Alias NYX1GW01 6

Network Name
 SPIFNET 1

LU Name NYX1GW01 5

5. Click the **Advanced** tab to display the window that is shown in the following figure:

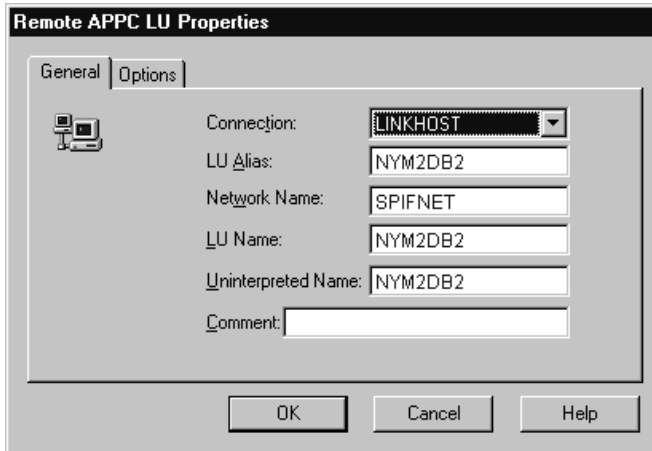


6. Activate the **Member of Default Outgoing Local APPC LU Pool** option by clicking its check box.
7. For **LU 6.2 Type**, click the **Independent** radio button.

8. Click **OK**.

Step 7: Setting Remote APPC LU Properties: To assign a remote APPC LU to a connection, complete the following steps:

1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **APPC**.
3. Select **Remote LU** to display the Remote APPC LU Properties window that is shown in the following figure. By default the **General** tab is displayed.

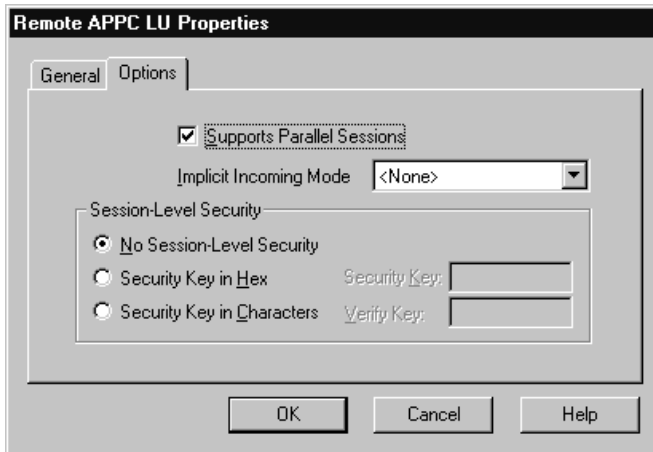


4. Use the values from "Windows NT Configuration Worksheet" on page 264, and "Windows NT Connection to Server Worksheet" on page 264. For example:

Connection	Select the connection you defined in "Step 4: Adding a Connection" on page 126 from the pulldown menu. In this example, it is LINKHOST.
LU Alias	NYM2DB2 11
Network name	SPIFNET 1
Remote LU name	NYM2DB2 11
Uninterpreted Name	NYM2DB2 11

When you select the value for **Connection**, the remaining values will be entered dynamically.

5. Click the **Options** tab to display the window that is shown in the following figure.



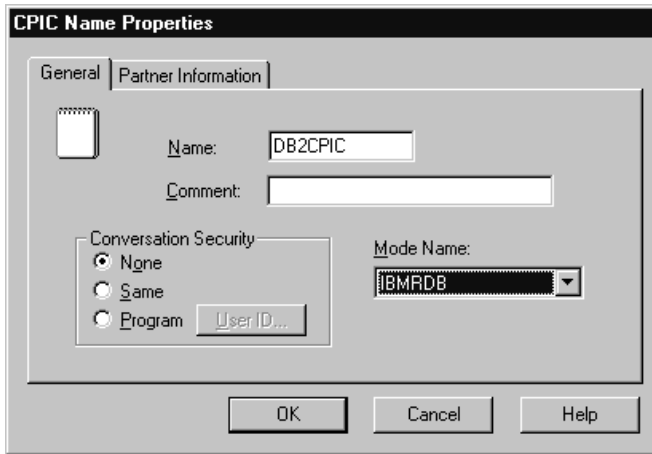
6. Make sure that the **Supports Parallel Sessions** check box is selected.
7. Click **OK**.

Step 8: Setting Symbolic Destination Name Properties: To configure the properties for CPI-C, complete the following steps:

DRDA AS consideration:

If configuring DataJoiner as a DRDA AS, you do not need to perform this step.

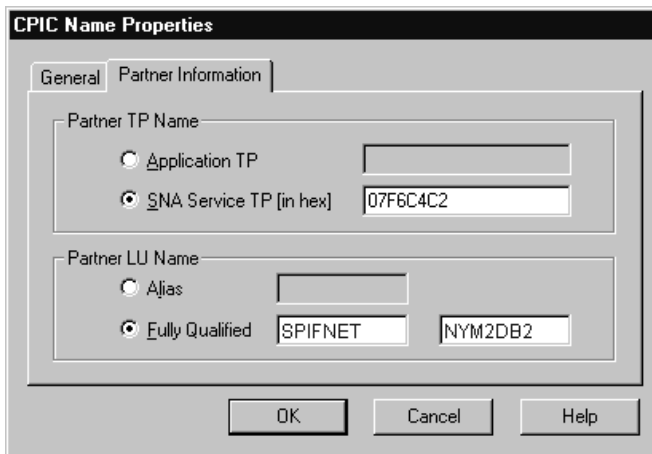
1. Select **Insert** from the **SNA Server Manager** menu bar.
2. Select **APPC**.
3. Select **CPIC Symbolic Name** to display the CPIC Name Properties window that is shown in the following figure. By default the **General** tab is displayed.



- Use the values from your worksheets, “Windows NT Connection to Server Worksheet” on page 264, and “Windows NT Configuration Worksheet” on page 264. For example:

Name	DB2CPIC 9
Conversation Security	By default, None is selected. You will specify the conversation security type later.
Mode Name	IBMRDB 4

- Click the **Partner Information** tab to display the following window:



- Refer again to the values from your worksheets, “Windows NT Connection to Server Worksheet” on page 264, and “Windows NT Configuration Worksheet” on page 264. For example:

SNA Service TP

07F6C4C2 12 (for MVS or OS/400)

Fully Qualified

SPIFNET.NYM2DB2 1.11

7. Click **OK**.

Step 9: Saving your Configuration: When you have configured your system and all of the server connections, save your configuration:

1. From the **File** menu, select **Save configuration**.
2. Activate your new configuration by stopping and starting Microsoft SNA Server by performing the following steps:
 - a. Select the service with which you want to work.
 - b. Select the **Services** menu, then **Stop Service** choice.
 - c. Select the **Services** menu, then **Start Service** choice.
3. Close the SNA Server Manager tool.

At this point, you need to return to “Chapter 7. Configuring Access to Data Sources” on page 51 to complete the configuration process for your data sources.

Configuring DataJoiner DDCS Using TCP/IP

This section shows you how to access DRDA data sources via TCP/IP.



Due to the characteristics of the TCP/IP protocol, the TCP/IP subsystem on one host might not be notified of the failure of its partner on a different host. As a result, a DataJoiner server accessing a data source using TCP/IP, or the corresponding process at the data source, might sometimes appear to be hung. You can use two network options, `tcp_keepidle` and `tcp_keepintvl`, to tune a specified time interval before the failure is detected.

The values for these options are system wide and apply to all TCP/IP socket applications for which the `SO_KEEPALIVE` socket option is enabled, not just DataJoiner.

To set or display the network options, use the `no` command.

This section can be used as a guide for configuring any DRDA data source that is accessed via TCP/IP.

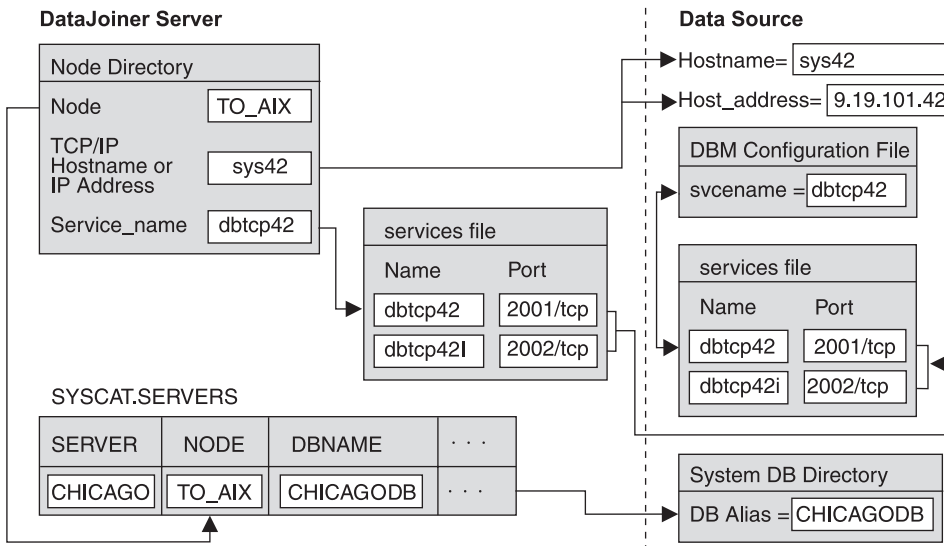


Figure 14. DRDA Data Source via TCP/IP

The values used in this section are in Figure 7 on page 68. Figure 8 on page 70 is an example of a blank template. You can fill in the blank template with your system's information and then complete the instructions in this section by plugging your own values in where appropriate.

On the data source system:

1. Create the database if you did not already. You will need this database's name or its alias to be able to complete the configuration needed on DataJoiner.
2. Make sure that each data source is installed and configured so it enables communication support for clients over TCP/IP. If this was not already done, see the data source's installation guide for instructions on how to do so.

You need two additional pieces of TCP/IP-related information before you can complete these instructions:

- The TCP/IP port number for the data source as it is stored in the TCP/IP services file on the data source's system. In the example, the entry would look like this:


```
dbtcp42      3700/tcp      # Data source primary connect port
```
- The host name of the system where the data source resides. To get this name, issue the hostname command from a command prompt on the data source's system. In the example, the hostname is sys42.

On the DataJoiner server workstation:

1. Update the TCP/IP services file that is typically found in C:\WINNT\system32\drivers\etc with the port number. All DRDA data sources communicate with TCP/IP clients through ports that are defined in the services file. Find the data source's port number in the system's services file. In the example, the entry would be:

```
dbtcp42      3700/tcp          # Data source primary connect port
```

2. Update the TCP/IP hosts file for the data source host, if necessary. Whether you must update the hosts file depends on how TCP/IP is configured on your network. Part of the network must translate the remote hostname that is specified on the **catalog tcpip node** command to an address. If your network has a name server that recognizes the host name, you do not need to update the hosts file. Otherwise, you need an entry in the hosts file for the remote host name. See your network administrator to learn how your network is configured.

At this point, you need to return to “Chapter 7. Configuring Access to Data Sources” on page 51 to complete the configuration process for your data sources.

Part 4. Using DataJoiner

Chapter 20. Performing Administration Tasks

This section describes how to perform administration tasks with DataJoiner's software tools. The topics covered include:

- Managing database directories with the Database Director. See *Managing Database Directories Using Database Director*.
- Tuning SQL statements using Visual Explain. See "Tuning SQL Statements Using Visual Explain" on page 141.
- Tuning system performance by using Performance Monitor's Snapshot Monitor and Event Monitor. See "Tuning Performance Using the DB2 Performance Monitor" on page 143.

Managing Database Directories Using Database Director

Use the Database Director's graphical interface to DataJoiner objects to perform the following tasks:

- Configure databases and database manager instances
- Manage the directories necessary for accessing local and remote databases and instances
- Backup and recover databases or table spaces.

Start the Database Director by double-clicking the **Database Director** icon in the **DataJoiner** folder.

Information about using the Database Director is covered in the Getting Started windows in the online help. To view the help, follow these steps:

1. Double-click the **Database Director** icon.
2. Select **Help**.
3. From the help pull-down menu, select **Getting Started**.

Tuning SQL Statements Using Visual Explain

Knowledge about how to best tune SQL queries is developed through experience over time. This information provides an overview on how to get started analyzing your SQL queries using Visual Explain. For detailed information, see the Getting Started information in the online help.

Visual Explain lets you view the access plan for explained SQL statements in graph format. You can then use the information available from the graph to find ways to tune your SQL queries for better performance.

The following steps enable you to create the examples as presented in the Getting Started information in the online help, and to perform basic tasks on the data.

1. Open the **DataJoiner** folder.
2. Double-click the **DB2 Command Window** icon, then enter **NET START DB2** at a command prompt to start the database manager for the DataJoiner instance.
3. Enter **DB2 CONNECT TO SAMPLE** at the command prompt to connect to the sample database. (If you have not yet created the sample database, follow the steps in “Creating the Sample Database and Selecting Sample Data” on page 35.)
4. Enter the following command in the drive:\SQLLIB\MISC directory:

```
DB2 -TF EXPLAIN.DDL
```

This command creates the explain tables that are based on sample queries.

5. Enter the following command in the drive:\SQLLIB\SAMPLES\VE directory:

```
DB2 -TF VESAMPL.DDL
```

This command imports the predefined snapshots that are based on sample queries.

6. Double-click the **Database Director** icon. The Database Director window opens.
7. Click the [+] for the **DataJoiner instance** where the SAMPLE database resides.
8. Click the [+] for **Databases**.
9. Click the [+] for **Sample**.
10. Double-click the **Explained Statements History** icon to view the explained statements.
11. Select an explained statement, click the **Statement** pull-down and choose **Show access plan** to view the graph of the explained statement. You can double-click each node in the graph to view statistics about the various components.
12. Click the **Statement** pull-down and choose **Show SQL text** to view the SQL statement that was explained. From here you can view the optimized SQL text to see how you can improve the performance of your queries.

Tuning Performance Using the DB2 Performance Monitor

Use the DB2 Performance Monitor to report on hundreds of DB2 performance attributes. Some of the common attributes include connections, sort activity, and I/O activity. You can view graphical summaries of the collected data or the complete details.

You can choose to monitor snapshots or events. Snapshot Monitoring allows you to capture point-in-time information at specified intervals. Event Monitoring allows you to record performance information over the duration of an event such as a connection.



You can also use the Windows NT Performance Monitor (PM) to display performance information that is retrieved from any of the performance data providers that are registered with the system. Windows NT provides performance data on all aspects of machine operation CPU usage, memory utilization, disk activity, and network activity. For information on how to use it, refer to the *DB2 Installation and Operation Guide*.

Snapshot Monitoring

Snapshot monitoring is useful for monitoring database objects such as databases, database manager instances, database connections, tables, and table spaces. For example, using the snapshot monitor can help you find information on the total number of connections.

The following steps enable you to create a snapshot of the SAMPLE database. Information will be captured for each of the performance attributes every 30 seconds. We will look at a graph of the results as well as the details.

1. At a command line, enter:

```
DB2 CONNECT TO SAMPLE
```

(If you have not yet created the sample database, enter the **db2sampl** command.)

2. Open the **DataJoiner** folder.
3. Double-click the **Database Director** icon. The Database Director window opens.
4. Click the **[+]** for the **DataJoiner instance** where the database resides.
5. Click the **[+]** for **Databases**.
6. Select **Sample** by clicking the icon.
7. Click the **Selected** pull-down and choose **Start monitoring** to begin monitoring the SAMPLE database. A standard set of information will be collected regarding the sample database.

8. From the Snapshot Monitor - Monitored Objects window, click the **Snapshot monitor** pull-down. Change the capture interval time to 30 seconds.
9. Click **OK**.
10. Enter the following SQL statement at a command prompt to generate some simple activity against the database:
DB2 SELECT * FROM EMPLOYEE ORDER BY LASTNAME
11. From the Snapshot Monitor - Monitored Objects window, click **SAMPLE**, then click the **Selected** pull-down. Choose **Open as** and then **Performance details** to view the standard information that is collected for a snapshot that was taken of the database.
12. Scroll through the information until you see the database connection information. Double-click the **Connection** entry to view detailed information on the database connections that were made. The number for **Total Connections** corresponds to the number of connections you made to the **SAMPLE** database.

Event Monitoring

Event monitoring gathers information about database events that occur during the running of an application. Event monitoring is useful for detecting deadlocks, overflow events, transaction completion, and application disconnections. For example, using the event monitor can help you find information on a slow-running application.

The following steps enable you to create an event and analyze the resultant data:

1. At a command line, enter **DB2 CONNECT TO SAMPLE**.
If you have not yet created the sample database, enter the command **db2sampl**.
2. Create an event monitor named **test1** that will collect data for statements and connections by typing the following command at a command prompt:
DB2 CREATE EVENT MONITOR TEST1 FOR CONNECTIONS, STATEMENTS WRITE TO FILE 'E:\SQLLIB\EV
3. Make sure that the directory you specify exists. If not, create it before activating the Event Monitor.
4. Activate the event monitor by typing the following command at a command prompt:
DB2 SET EVENT MONITOR TEST1 STATE=1
5. Generate activity against the database by opening several windows, connecting to the **SAMPLE** database, and issuing simple queries such as:
DB2 SELECT * FROM EMPLOYEE
6. Turn off the event monitor by issuing the following command:
DB2 SET EVENT MONITOR TEST1 STATE=0

This forces the event monitor to write the trace file. If not turned off, a write occurs only once the buffer is full.

7. Activate the Event Analyzer to look at the trace information that is collected by performing the following steps:

- a. Type the following command in the DB2 command window:

```
DB2EVA -PATH <drive:>\SQLLIB\EVENTS
```

The Monitored Periods View opens.

- b. Select a **monitored period**, click the **Selected** pull-down. Choose **Open as** and then **Connections**. This shows the number of connections you made.
- c. Click the **Selected** pull-down and choose **Open as** and then **Statements**. This provides information on the SQL statements you issued during the connection.

As you go through this exercise, you will see many other options. Feel free to explore any of them. Read the *Getting Started* online help to see detailed information on each of the options. The online help also provides information on monitoring and analyzing your own applications.

Chapter 21. Setting Environment Variables

Environment Variables control your database environment.

During the DataJoiner installation, the program adds or updates the following System Environment Variables:

- **DB2PATH:** The value for DB2PATH is set to %DB2PATH%, the install path.
- **PATH:** The %DB2PATH%\BIN and %DB2PATH%\FUNCTION are added to the existing path.
- **INCLUDE:** The %DB2PATH%\INCLUDE is added to the Include variable.
- **LIB:** The %DB2PATH%\LIB is added to the Lib variable.
- **DB2INSTANCE:** The value for DB2INSTANCE is set to DB2.
- **BOOKSHELF:** The value for BOOKSHELF is set to %DB2PATH%\BOOK.
- **HELP:** The value for HELP is set to %DB2PATH%\HELP.

You can determine the setting of environment variables by using the **echo** command. For example, to check the value of the DB2COMM environment variable, enter:

```
echo %db2comm%
```

You can set environment variables as needed, or alter the values that are set by **setup** by updating the Windows NT registry as follows:

1. Double-click the **Program Manager** icon.
2. Double-click the **Main** icon.
3. Double-click the **Control Panel** icon.
4. Double-click the **System** icon.
5. In the **System Control Panel**, in the **System Environment Variables** section, do the following:
 - If the environment variable does not exist, do the following to add it:
 - Select any environment variable.
 - Change the name in the **Variable:** section to the environment variable to be added; for example, DB2COMM.
 - Change the **Value:** as required; for example, APPC,TCPIP.
 - If an environment variable already exists, do the following to append a new value:
 - Select the environment **Variable:** to be updated; for example, DB2COMM.
 - Change the **Value:** as required, separated by commas; for example, APPC,TCPIP.

6. Select **Set**. The environment variable is added in alphabetical order, if it did not exist.
7. Select **OK** to exit.

You also can update environment variables by using a command prompt or running command files that you create for this purpose. For example:

```
set db2instance=inst2
```

This command makes `inst2` the active instance for the Windows session that issues it.

```
set db2comm=tcpip
```

This command makes `tcpip` active for the Windows session that issues it.

DataJoiner Environment Variables

The following is a list of DataJoiner environment variables with a brief description of each; some may not apply to your environment.

Environment variable requirements:

All environment variables that are intended for DataJoiner services must be defined as **System Environment Variables**. When adding or updating System Environment Variables, your system must be rebooted before the new values are visible to DataJoiner services.

Performance

You can control whether a data access module is loaded at DataJoiner initialization through the use of the `DJXCOMM` environment variable. By default, DataJoiner loads a data access module each time a data source is accessed. `DJXCOMM` tells DataJoiner to load a data access module when DataJoiner is started. Preloading data access modules can improve performance because the data access module is loaded only once between the time DataJoiner is started and stopped.

DJXCOMM Default = null (OFF). Value: *data-access-module*.

Valid values include `drda`, `db2ra`, `sqlnet`, `informix`, and any others that are allowed in the `SERVER_PROTOCOL` column of `SYSCAT.SERVERS`.

Multiple data access modules can be loaded using `DJXCOMM`.

General

DB2ACCOUNT

DDCS accounting string.

DB2CHKPTR [Default=null (OFF), any value (ON)]

Selectively turns pointer checking ON for debugging; it should be left OFF for normal operation.

DB2CODEPAGE [Default=null (use system codepage), values: (valid codepage)]

Used to override the system code page.

DB2CODEPAGE needs to be used when talking to servers that do not understand the default codepage from the client. This situation occurs when attempting to communicate with downlevel DataJoiner and DB2 servers and DDCS host databases. DB2CODEPAGE is used to set the default codepage to a codepage that the server supports.

For example, Windows NT Latin 1 codepage 1252 is not supported on DB2 for OS/2 Version 1.2 servers. To connect to a DB2 for OS/2 Version 1.2 server from DataJoiner for Windows NT, you could set DB2CODEPAGE=437.

See “Country Code and Code Page Support” on page 271 for a list of compatible codepages for DataJoiner for Windows NT.

DB2DBMSADDR [Default=null, value: (9000000)]

On a Windows NT system, you need to override the default value (2000000) only if you get an error message that indicates that you could not create database-shared memory.

DB2DMNBCKCTLR [Default=null, values:(null, ?, <DOMAINNAME>)]

Specifies which machine is the backup domain controller:

- **null** causes DataJoiner to determine the machine upon which the user is known; this will be either the local machine or the primary domain controller that knows the user. All subsequent requests are sent to that machine.
- **?** causes DataJoiner to determine whether the local machine is a backup domain controller or a primary domain controller.

- If it is neither, security checks are made as for **null**.
- If it is either type of domain controller, this value (?) confirms that the domain for which the user is known is equal to the domain for which the local machine is a controller. If this is not the case, all subsequent requests are sent to the primary domain controller.
- *<DOMAINNAME>* (a domain name in upper case) causes DataJoiner to confirm that the local machine is a backup domain controller:
 - If it is, requests are handled as for ?
 - If it is not, a message that describes the local machine's actual attributes is displayed.

DB2DBDFT [Default: SAMPLE]

Specifies the database alias name of the database that will be implicitly connected to when applications are started.

DB2INSTANCE

Specifies the instance that is active by default.

DB2INSTPROF

Specifies the location of the instance directory, if different than %DB2PATH%.

DB2LOADREC

Specifies the "load copy location file"; it is used for load recovery.

%DB2PATH%

Specifies the directory where the product is installed.

Command Line Processor

The following is a list of DataJoiner for Windows NT environment variables that affect the command line processor.

DB2BQTIME [Default= 1 second, Minimum value= 1 second]

Specifies the amount of time the command line processor front end will wait before checking if the back end process is active and establishing a connection to the process.

DB2BQTRY [Default= 60 retries, Minimum values= 0 retries]

Specifies the number of times the command line processor front end process tries to

determine whether the back end process is already active. It works in conjunction with DB2BQTIME.

DB2RQTIME [Default= 5 seconds, Minimum value= 1 second]

Specifies the amount of time the command line processor back end process waits for a request from the front end process.

DB2IQTIME [Default= 5 seconds, Minimum value= 1 second]

Specifies the amount of time the command line processor back end process waits on the input queue for the front end process to pass commands.

DB2OPTIONS

Specifies command line processor options.

Communications

Environment variable requirement:

When running DataJoiner for Windows NT as a service, the following environment variables must be present in the System Environment Variables within the Control Panel at system startup (to take effect). The settings of these environment variables would then apply to all instances.

DB2COMM [Default= null, values: TCPIP APPC IPXSPX NETBIOS NPIPE]

Specifies which communication protocols will be enabled when the database manager is started; it may be set to any combination of the values that are listed, separated by commas.

DB2NBADAPTERS

Specifies which adapters to use for DataJoiner NetBIOS LAN communications; it allows the selection of multiple adapters. Each adapter is specified using its logical adapter number.

DB2NBSESSIONS

Specifies the number of NetBIOS sessions to be reserved for DataJoiner; it allows specific session requests for each adapter that is specified in the DB2NBADAPTERS variable.

DB2NBINTRLISTENS

Specifies the number of NetBIOS listen NetBIOS control blocks (ncbs) that will be asynchronously issued in readiness for remote client interrupts; it allows specific session requests for each adapter that is specified in the DB2NBADAPTERS variable. This flexibility is provided for "interrupt active"

environments, Specifies the number to ensure that the interrupt calls from remote clients will be able to establish connections when servers are busy servicing other remote interrupts.

DB2NBRECVNCBS

Specifies the number of NetBIOS “receive_any” commands (ncbs) that the server will issue and maintain during operation; it allows specific session requests for each adapter that is specified in the DB2NBADAPTERS variable. This value may be adjusted depending on the number of remote clients your server is connected to. Lower values will conserve server resources. Higher values may be required to prevent clients from retrying sends (when high traffic results in all receive_any commands being busy).

DB2NBSENDNCBS

Specifies the number of send commands (ncbs) that the server will reserve for use. This value may be adjusted depending on the number of remote clients that are connecting to your server. Lower values will conserve server resources. Higher values may be required to prevent the server from waiting to send to a remote client when all other send commands are in use.

DB2NBRECVBUFFSIZE

Specifies the size of the DataJoiner NetBIOS protocol receive buffers. These buffers are assigned to the NetBIOS receive ncbs.

DB2NBCHECKUPTIME

Specifies the time interval between each invocation of the NetBIOS protocol checkup procedure. Checkup time is specified in minutes.

DB2SERVICETPINSTANCE

Specifies the name of the instance that supports *down level* clients using the APPC protocol.

ADSM

There are three environment variables that are related to ADSM:

- DSMI_CONFIG
- DSMI_DIR
- DSMI_LOG

See the *DB2 Administration Guide* for more information about ADSM and these three variables.

Part 5. Replicating Data

This section describes how to plan, configure, administer, and operate DataJoiner's replication administration tool. Because the replication administration tool has many of the same capabilities as the IBM DB2 Universal Database Replication Version 5 product,² this section merely expands the scope of the *IBM DB2 Replication Guide and Reference (V5)*, focusing on replication that involves non-IBM databases. All information about non-IBM replication can be found in this book. Information about replication between IBM's DB2 databases is documented in *IBM DB2 Replication Guide and Reference (V5)*. We suggest you read through *IBM DB2 Replication Guide and Reference (V5)* to understand the basics of IBM replication.

This section uses industry standard terminology for database, copying, and LAN concepts, except where noted otherwise. A comprehensive glossary is located in the *DB2 Universal Database Glossary*. The following terms represent concepts specific to the replication administration tool:

Control server

The database where the subscription definitions are located. The *control server* can be co-located at either the source or target server or at a DB2 database that is neither the source nor the target. The control server is usually located where the Apply program runs because the Apply program frequently reads the tables in the control server.

Differential-refresh copying

In differential-refresh copying, only the data that has changed in the source table since the last replication cycle is copied to the target table.

Full-refresh copying

During full-refresh copying, the Apply program:

1. Deletes all the rows from the target or target subset view
2. Reads all the rows from the source table or view
3. Copies (*refreshes*) the source rows to the target table.

Non-IBM server

A relational database that is not an IBM produced product. The non-IBM database (for example, Oracle, Sybase, or Microsoft SQL Server) is identified (or defined) to DataJoiner to be accessed for replication. When a DataJoiner database is identified as a source or

2. Sometimes referred to as V5 Apply and V5 Capture programs.

target server, the heterogeneous server must be identified to DataJoiner, specifying the specific Oracle, Sybase, Microsoft SQL Server, or other data source.

Replication

Refers to the propagation of data from source tables to target tables. *Heterogeneous replication* is replication that involves a non-IBM database that is either a source or target for replication.

Sources

Source tables or DB2 source views in a replication scenario.

Source server

When the source data is in DB2, the source server is the database where the source data is stored and where the Capture program runs. When the source data is in a non-IBM database, the source server is a DataJoiner database and Capture triggers are located on the non-IBM database that holds the source data. The non-IBM database is accessed through DataJoiner.

Subscription members

A specification that identifies (to the Apply program) the exact source table that contains the data to be copied and the exact target table that the data will be copied to. This specification is one item in a subscription set.

Subscription set

A set of subscription members and their attributes that are all processed together in a single Apply program unit of work (UOW). Common attributes include: source server, target server and Apply qualifier.

Target server

The database where the target copy of the data is stored. When the target data is in a non-IBM database, the target server is a DataJoiner database.

Update-anywhere replication

Replication in which the replication source or the target table can be updated and the changes replicated to all related tables. Transaction-conflict detection is available only for replication between DB2 family sources and targets. Row-level conflict detection is supported between Microsoft Jet and any of the following: Oracle, Sybase, Microsoft SQL Server, and Informix.

Chapter 22. Data Replication—What is it?

Basically, data replication consists of two processes:

- The capturing of changed data from a table or view in a *source* database
- The copying (or “applying”) of changed data from a *source* table to one or more *target* tables in the same or different databases

Although some aspects can vary, the basic processes are change capture and data copy. The IBM Replication Solution provides flexibility in determining how the changed data is to be captured and how (and under what conditions) this data is to be copied to the targets.

You must administer the two basic processes. Replication administration is the process of creating and managing your replication environment. The administrative tasks consist of:

- Defining a table or view as a source table for change capture and replication
- Creating a target table
- Defining the conditions under which the data from the defined source table is copied to the target table

Additional administrative tasks include preparing a database server to participate in replication or changing an existing source table.

The IBM Data Replication Solution

IBM’s data replication solution consists of data replication tools and products that function together in a single solution. The tools and products are:

- DataJoiner’s replication administration tool: The tool for administering replication between IBM and non-IBM databases. The replication administration tool works with DataJoiner, Capture triggers, and the Capture and Apply programs.
- DB2 Universal Database (DB2 UDB): The Capture and Apply programs packaged³ with DB2 UDB are compatible with DataJoiner’s replication administration tool and DataJoiner V2.
- DataPropagator Relational (DPROPR) Capture and Apply programs
- DB2 DPROP for Microsoft Jet
- IMS DataPropagator (formerly called DataPropagator NonRelational)
- DataRefresher

- DataJoiner
- Lotus NotesPump

This is an overview of how these products support the complete data replication solution. The discussion here is about the replication features between IBM and non-IBM databases only. DB2 UDB V5, DPROPR Capture and Apply programs, DPROPNR, DataRefresher, DataJoiner, and Lotus NotesPump are discussed in other IBM publications.

Support of Replication Environments

By supporting sources and targets that include the DB2 family, IMS, VSAM, Oracle**, Sybase**, Microsoft**, Informix**, Lotus Notes, and others, IBM's replication solution ensures that you have timely, reliable, and consistent data across your enterprise. IBM supports replication in heterogeneous environments as follows:

- IBM's architecture is built on standard SQL to leverage the database capabilities for data enhancement, network connectivity, and data security.
- An architected data staging area supports interoperability among multivendor databases, between heterogeneous data models, and among products from independent software vendors. IBM's DPROPNR, working with DataRefresher,⁴ brings IMS, VSAM, and flat file data into the staging area, making it available for replication to client/server targets.⁵ Additionally, independent vendors offer products that read and write the data staging tables relying on IBM Capture programs, DataJoiner, and the Apply program.
- IBM replication products directly support multivendor sources and targets through IBM's multidatabase server product, DataJoiner V2. DataJoiner provides transparent access to Oracle, Sybase, Microsoft SQL Server**, and Informix, thus enabling replication to run across multivendor environments.
- Lotus NotesPump extends replication to Lotus Notes and Open Database Connectivity (ODBC)-accessible data (for example, data that can be accessed using Lotus Approach or Microsoft Access).
- DB2 DPROPR for Microsoft Jet extends IBM's enterprise data replication solution to support Microsoft Access and Microsoft Jet databases in LAN, occasionally-connected, and mobile environments.

3. The Capture and Apply programs are separately-priced features of DB2 UDB for OS/390. They are packaged with DB2 UDB on Windows 9x and NT, OS/2, and on AIX and other UNIX platforms.

4. DataRefresher and ETI•Extract, by themselves, provide full-refresh capabilities. When used with the Data Difference utility (DDU), they can provide differential-refresh capabilities. The DDU is packaged with the latest versions of IBM's DPROPNR and Apply products.

5. See <http://www.software.ibm.com/data/dproprn/> on the Internet for more information about DPROPNR and DataRefresher.

Replication Solution and Product Environment

Figure 15 conceptually shows the relationships between the components of IBM's replication solution.

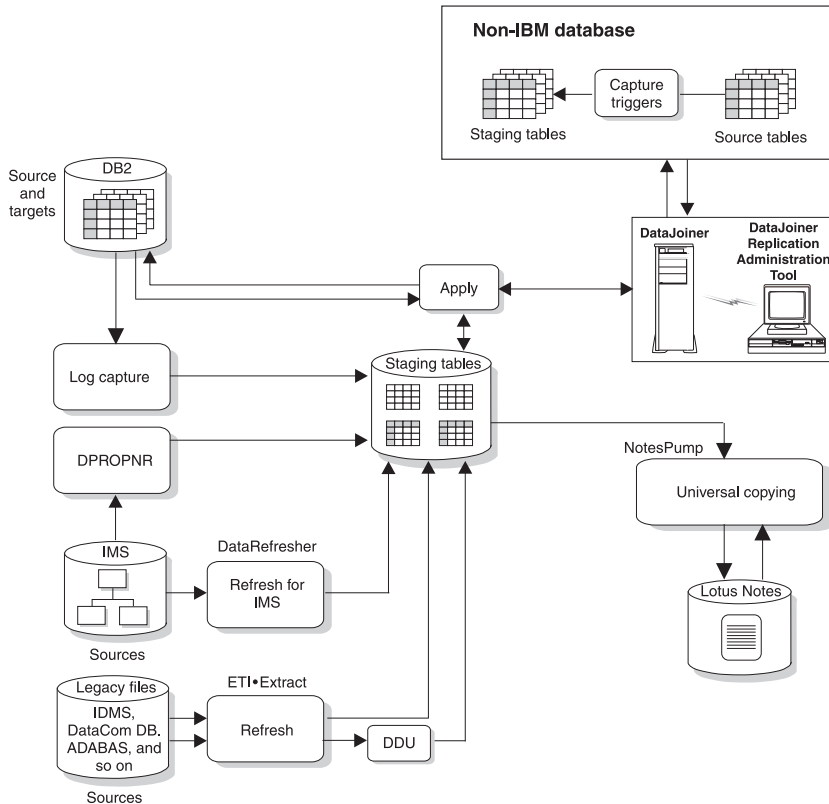


Figure 15. IBM's Data Replication Solution and Product Environment

For non-IBM sources, the replication administration tool works through DataJoiner to create Capture triggers on the source tables. These triggers capture the changed data for copying and place it into staging tables on the non-IBM source server. The Apply program accesses the source server through DataJoiner nicknames. The updates can be subsequently staged at any number of sites, allowing for flexible data distribution.

For non-IBM targets, the replication administration tool works through DataJoiner to create the target tables on the non-IBM server and nicknames for those target tables in a DataJoiner database. The Apply program then

applies (copies) the changed data destined for the targets to the nicknames and DataJoiner applies those changes to the real target tables on the non-IBM server.

The Apply program can filter, enhance, distribute, and replicate data from the source tables to the target tables. DataJoiner enables the Apply program to *transparently* read from or write to the non-IBM sources and targets.

Multivendor Data Sources and Targets

IBM's DataJoiner enables replication between IBM and non-IBM sources and targets, including DB2, Oracle, Sybase, SQL Anywhere, Microsoft SQL Server, and Informix. DataJoiner can help your business meet the challenge of efficiently accessing distributed data by enabling users to develop a virtual, enterprise-wide relational database.

DataJoiner provides a single-site, local image of all your data, relational and nonrelational, local and remote, from IBM and non-IBM platforms. DataJoiner masks the differences in SQL dialects, data access methods, networking protocols, and operating systems. In the context of replication, DataJoiner, together with the replication administration tool, provides a configuration that can access multivendor sources or targets as if they were all a DB2 database.

Chapter 23. The Replication Tools

The tools that provide support for the IBM replication solution are:

DataJoiner's replication administration tool	The replication administration tool serves as your replication administration interface. This tool runs on Windows NT or 9x.
DataJoiner	DataJoiner provides access to non-IBM databases for replication purposes.
Capture program	The Capture program captures the data changes from DB2 databases for replication purposes.
Capture triggers	The Capture triggers capture the data changes from non-IBM databases for replication purposes.
Apply program	The Apply program performs the replication of the data from the source database to the target database.

The following sections describe how the replication administration tool, DataJoiner, the Capture program, the Capture triggers, and the Apply program work and interact with each other.

The Replication Administration Tool

The replication administration tool serves as your replication administration interface, automating many replication activities. From the replication administration tool, you can select your source tables and prepare them for change capture, create your target tables, and create the control information for the Apply program. The control information causes the Apply program to copy data from the selected source tables to the selected target tables. You can also use the replication administration tool to browse or change replication activities.

The databases that the replication administration tool supports as sources or targets are:

- DB2 UDB V5 and V6
- DB2 for MVS V3R1 or V4R1, and DB2 for OS/390 V5R1 or V6
- DB2 for common servers V2 and DataJoiner V2
- Oracle V7.0.13
- Informix V7.1 or later
- Sybase V4R6 or later (UNIX) and Sybase V11 (Windows NT)
- Sybase SQL Anywhere Version 5.0 (Windows NT)
- Microsoft SQL Server V4.21 or later (UNIX) and Microsoft SQL Server V6.0 or later (Windows NT)
- DB2 DPROP for Microsoft Jet (as target only)

You can use DataJoiner with the replication administration tool to define, synchronize, automate, and manage copy operations from a single control point for data across your enterprise, as shown in Figure 16. You can also tailor or enhance data as it is copied, thus delivering detailed, divided, summarized, or derived data when and where it is needed.

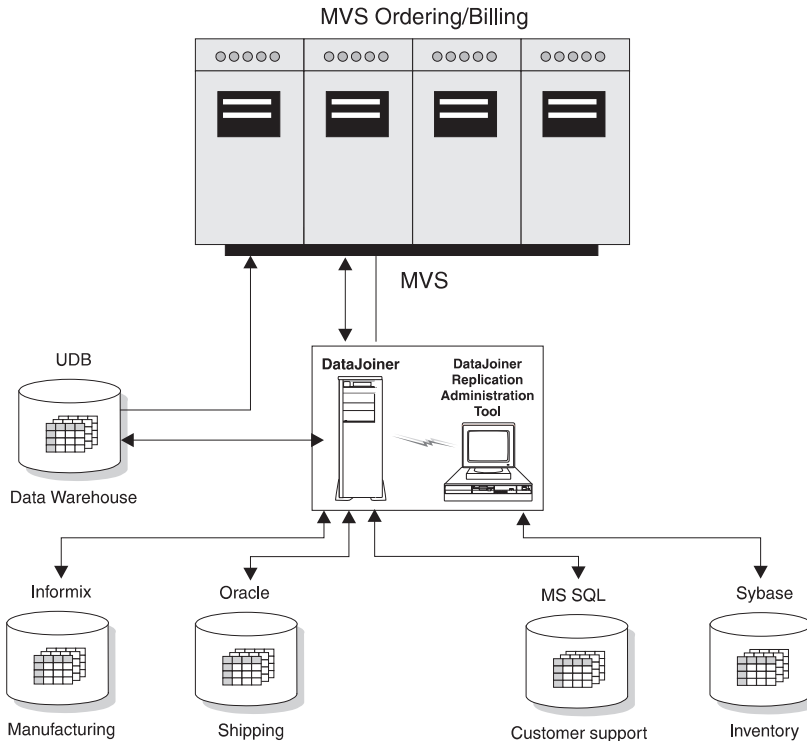


Figure 16. DataJoiner with the replication administration tool Scenario. DataJoiner with the replication administration tool enables you to copy data across your enterprise.

The replication administration tool provides objects and actions that define and manage source and target table definitions. Working through DataJoiner, the replication administration tool create:

- Capture triggers on the non-IBM source servers
- Nicknames in the DataJoiner database for the remote tables where the changed data is to be captured
- Target tables (and their associated nicknames) in the non-IBM database for the remote target tables

The Apply program then reads from and writes to DataJoiner nicknames, eliminating the need to connect explicitly to non-IBM databases.

If the source database is a DB2 database, the Capture program for that database captures the changes, therefore, the Capture triggers and DataJoiner are not involved. If the target database is a DB2 database, the Apply program writes the changed data to the DB2 target database directly and DataJoiner is not involved.

Administering Replication: Overview

When you use the replication administration tool to perform replication administration tasks, the replication administration tool connects to the source, target, or control server in order to create and update the control information and target tables on the server (depending on the operation performed). The client workstation where the replication administration tool is located, must be authorized and able to connect to all the source, target, and control servers that are managed by the replication administration tool.

For DB2 source, target, or control servers, DataJoiner's distributed database connection services (DDCS) or the IBM DDCS product provides connectivity.

For non-IBM sources and targets, the replication administration tool uses DataJoiner to connect to the non-IBM servers. Non-IBM databases are not supported as control servers.

The replication administration tool graphical user interface (GUI) is divided into five areas that deal with control tables, promoting tables and subscription sets, sources, subscription sets, and the running or editing of SQL (see Figure 17 on page 164).

Using the GUI, you can perform the following administration tasks:

- Define DB2 tables, non-IBM tables, and DB2 views as sources. Sources are the tables or views that data is copied from.
- Create the control tables and put them on your source, target, and control servers. The control information for the replication process is kept in these control tables.
- Change the definitions for existing DB2 source tables in order to add new columns
- Remove or drop source tables and servers
- Define target table definitions, which consist of two parts:
 - Subscription members, which are requests for copying data from one source table to one target table
 - Subscription sets, which contain and group your subscription members into a single unit of work
- Change existing subscription members for DB2 target tables in order to add new columns

- Remove or drop subscription sets or subscription members that are no longer needed
- Add SQL statements or delete SQL statements or CALLED procedures that should be run before or after the target tables are replicated
- Run or edit SQL that is generated by the replication administration tool.

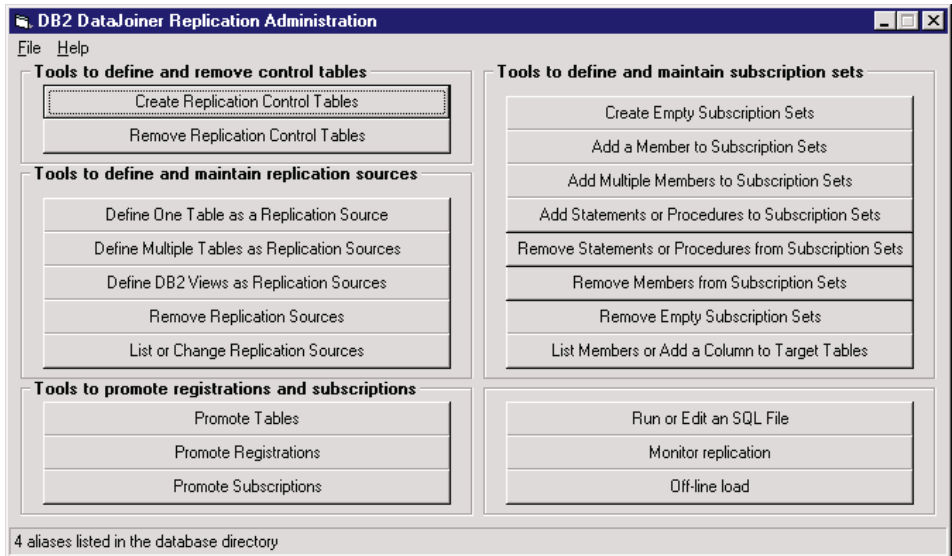


Figure 17. The Replication Administration Tool's Primary Window. The replication administration tool is divided into five areas: control tables, sources, promote, subscription sets, and SQL.

How the Replication Administration Tool Manages the Replication Environment

The replication administration tool helps you perform three major tasks:

- Creating or removing control tables
- Defining, maintaining, or removing source definitions
- Creating or removing subscription sets or subscription members

Source tables are tables that data is copied from. The copy of the data can then be updated or inserted into, or deleted from, one or more target tables.

The Control Tables

This section contains an overview of the control tables. The details of each are discussed in “Tables Used by the Capture Triggers” on page 171, “Control Tables Used by the Apply Program” on page 174, the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*.

Various components of the IBM replication solution use a set of control tables to hold information about:

- The source tables that are defined on a given source server
- The subscription sets that have been created between a specific set of source tables and target tables
- The information that is needed for communication between the Capture program (or Capture triggers) and the Apply program
- The information that is needed to manage (prune) the size of staging tables to prevent them from growing indefinitely

Not all of these control tables are used at every server. However, control tables that are used on source servers *must* exist in any DB2 or DataJoiner database to which the Apply program is bound (the Apply program contains static SQL that accesses those tables). For this reason, IBM recommends (and the replication administration tool generates) SQL to create all of the control tables on any specified source, target, or control server.

In order for the replication administration tool to put the control tables on the specified servers, the replication administration tool must use user IDs and passwords that have the proper authorized access to those servers. You define the user IDs and passwords for particular servers by using the replication administration tool. See “Setting Administrative Preferences” on page 219 for more information about specifying these user IDs and passwords.

Clicking **Create Replication Control Tables** from the replication administration tool primary window creates the control tables. This action generates SQL that can run immediately, can be edited (for customization purposes), or can run at a later time.

Note: You *must* create the control tables (that is, run the generated SQL) on a server *before* performing any replication requests (such as defining sources or creating subscription sets) that affect that server.

Sources and Targets

You can create your source and subscription member definitions through the replication administration tool. You use three objects to set up and maintain your replication environment:

Source definitions

Identify data in the source table as data to be replicated. This information is stored in the register control table.

Subscription member

Contains information that is used for copying data from one source table to one target table.

Subscription sets

Contains subscription members. The subscription sets group subscription members that are associated with a particular Apply program instance into a single unit of work. Subscription sets also describe the relationships between the source tables and one or more targets and contain the specifications for the target tables, their location, structure, a timing schedule, and, if the data being replicated is to be manipulated, the SQL necessary to do so.

SQL Generated for Replication Requests

Most of the replication administration tool functions generate SQL which, when run, creates, updates, or drops the specified objects on your servers.

When you choose the replication administration tool function, you can edit, save, and run the generated SQL immediately, or you can save the generated SQL and run it later. The file is saved as an ASCII file in your workstation file system.

If you choose to save the generated SQL as an ASCII file, you can edit it and run it at a later time through the replication administration tool. See “General Steps for Setting up Replication” on page 228 and “Editing the Replication Administration Tool-Generated SQL” on page 229 for more information about generating and editing SQL by using the replication administration tool. See “Running the Replication Administration Tool-Generated SQL” on page 230 for more information about running the SQL.

If you defer running the generated SQL, you can customize the replication tasks for your shop or application and determine when and how you run the SQL. You defer the processing of the replication action until a specified time, and create libraries of files containing SQL for backup, or site-specific customizing. You can also rerun the definitions as necessary. See “General Steps for Setting up Replication” on page 228 for more information on deferring and customizing generated SQL.

Whether you run the generated SQL immediately or at a later time, you must perform the activities for generating and running the SQL in a particular order. The order ensures that you create control tables at the source server before defining replication sources and that you create control tables at the source server *and* the control server before creating replication subscriptions. A recommended order is:

1. Generate SQL to create control tables at the source server and the control server.
2. Tailor the generated SQL if necessary.
3. Run the SQL to create control tables at the source server.

4. Generate SQL to define replication sources at the source server. The replication administration tool assumes that the source tables, nicknames, or DB2 views that are to be defined as a replication source already exist.
5. Run the generated SQL to define the replication sources at the source server.
6. Run the SQL to create control tables at the control server.
7. Generate SQL to create subscription sets on the control server.
8. Run the SQL to create subscription sets on the control server.
9. Generate the SQL to create subscription members at the control server.
10. Run the SQL to create subscription members at the control server.

DataJoiner

The replication administration tool accesses non-IBM databases through DataJoiner. DataJoiner presents a single view for various DB2 family and non-IBM databases, masking differences in data types, SQL dialects, and communications. You create nicknames for the non-IBM source or target tables in a DataJoiner database and then use the replication administration tool to create a source definition for the nicknamed object. Within a single DataJoiner local database, you can define the nicknames for one or more source tables that reside on *one* remote, non-IBM source server.

After you create subscription sets (using the replication administration tool) for a non-IBM source server, the Apply program connects to the DataJoiner database that is associated with the non-IBM server and accesses (through

nicknames) the information in the register control table and the staging table on the non-IBM source server (see Figure 18).

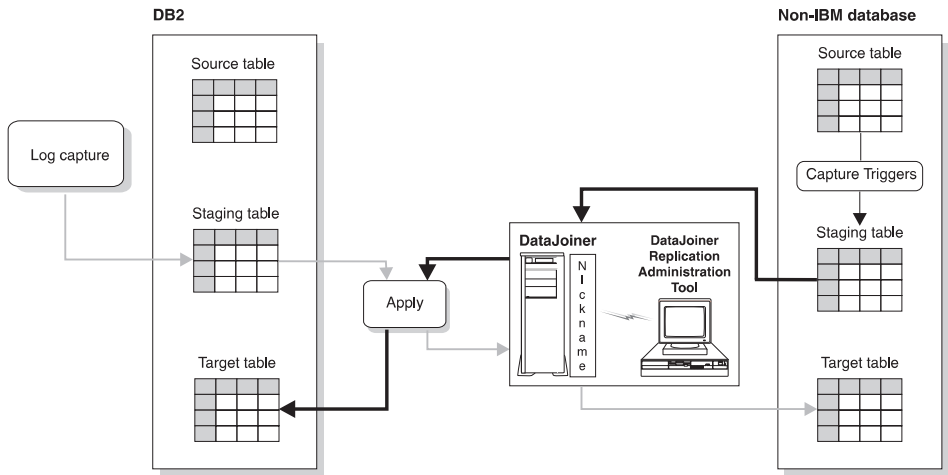


Figure 18. *DataJoiner in Action.* In a scenario where the source table is a non-IBM table (the dark arrows), DataJoiner nicknames give the replication administration tool and the Apply program access to the non-IBM source server and to changes made to the non-IBM source table (through the staging table). In a scenario where the source table is a DB2 table (the light arrows), DataJoiner nicknames give the Apply program access to the non-IBM target tables.

The Capture Program for DB2 Sources

The Capture program is the replication tool that captures the changed data from a DB2 source table (through DB2 logs) and makes the changed data available for replication. The Capture program runs at the DB2 source server database. Refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for more information on the Capture program.

Capture Triggers for Non-IBM Sources

Capture triggers are used for replication from non-IBM databases. They capture changed data from a source table and make the changed data available for replication. Capture triggers perform the same task as the Capture program does for DB2, but in a different manner. The replication administration tool generates the Capture triggers; they run in the source server database.

Capture Triggers Overview

The replication administration tool, working through DataJoiner, creates Capture triggers at the non-IBM source database when you define that

database as a source server. Capture triggers capture committed changes made to source data and places the captured changes into a staging table, called the *consistent change data* (CCD) table. The CCD table has a nickname in DataJoiner that programs that want to replicate the changes (for example, the Apply program) can access. See “Changed Data Tables” on page 181 for more information about CCD tables.

There are three triggers for each source table: DELETE, UPDATE, and INSERT.

How the Capture Triggers Capture the Data Changes

The Capture triggers work with three objects: the CCD table, the register control table, and the pruning control table. The pruning control table is discussed in “Pruning the CD and CCD Tables” on page 183.

The replication administration tool generates SQL that, when run:

- Creates Capture triggers on the source table
- Creates the CCD table on the source server
- Inserts a row into the register control table (to represent the new source table)
- Creates a nickname for the CCD table in the DataJoiner database

Whenever a delete, update, or insert operation occurs at the defined source, a Capture trigger records the change into the CCD table and puts a marker in the register control table. When the Capture triggers retrieve changed information, they can also obtain before and after columns data to put into the CCD table.

The Apply program looks at a marker in the register control table to identify changes that have not yet been replicated. The Apply program then reads the CCD table (through DataJoiner nicknames), copies the changes to the target server, and applies the changes to the target table. Figure 19 on page 170 shows the relationship between the Capture triggers, the source table, the register control table, and the CCD table.

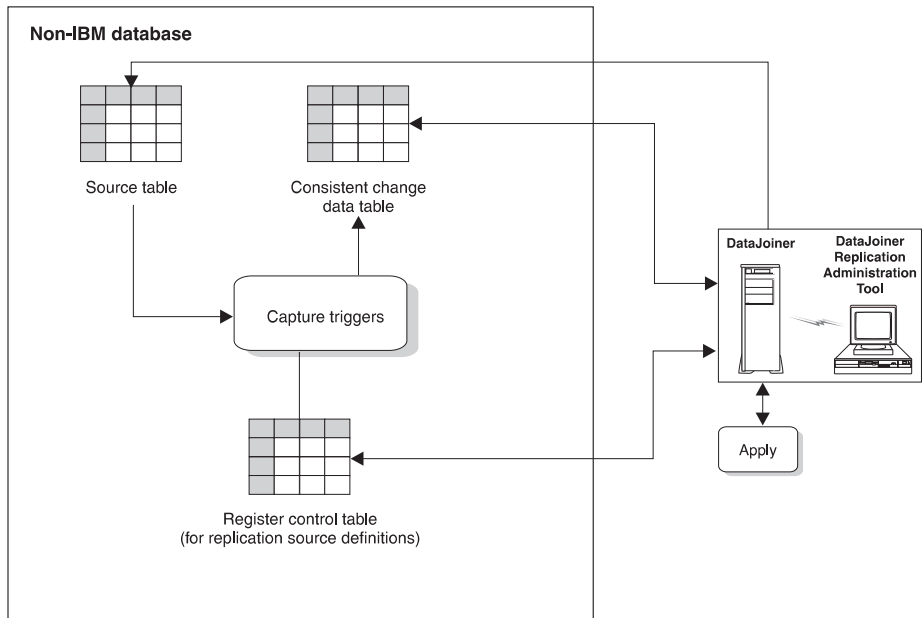


Figure 19. Capture Triggers at the Source Server. The Capture triggers monitor source changes, capture the changed data, and write the changed data to the CCD table.

Capture Triggers and Pre-existing Triggers

When the replication administration tool creates and places Capture triggers on a non-IBM database, you may experience the following occurrences:

- On Oracle: If an attempt is made to create a Capture trigger on a table where there is a pre-existing trigger with the same name, or where the pre-existing trigger performs an identical event (insert-before, insert-after, delete-before, delete-after, update-before, update-after), Oracle issues the following message: ORA-04081 (trigger name already exists). If this error is generated, the Capture trigger is not created.
- On Informix: If an attempt is made to create a Capture trigger on a table where there is a pre-existing trigger with the same name, or where the pre-existing trigger performs an identical event (insert, delete, update), Informix issues an -741 error and will not create the Capture trigger.
- On Microsoft SQL Server or Sybase: If an attempt is made to create a Capture trigger on a table where there is a pre-existing trigger with the same name, or where the pre-existing trigger performs an identical event (insert, delete, update), Microsoft SQL Server and Sybase do not generate error or warning messages indicating a conflict. Microsoft SQL Server and Sybase replace the pre-existing trigger with the new Capture trigger. However, the replication administration tool does check to see if a trigger already exists. If a trigger with the same events exists, the replication

administration tool creates the new triggers but all lines within the trigger body are commented out. You must determine how you want to merge pre-existing triggers with the new triggers. Then you can uncomment out the lines in the new triggers.

If you anticipate conflict between the replication administration tool's Capture triggers and pre-existing triggers, we recommend that you put the content of both triggers into one trigger. For each table event, append the pre-existing business trigger to the end of the Capture trigger script that is generated by the replication administration tool.

Tables Used by the Capture Triggers

Two control tables and one staging table are used on each non-IBM source server to manage the task of capturing changes to the data in the source table. These tables (and, in the case of the Apply program, their nicknames in the DataJoiner database) are accessed and updated by the Capture triggers and the Apply program to keep track of:

- Capturing changed data for the source tables that are defined on that source server
- Applying the changed data to the various target tables

You must create the control tables on the source server (using the replication administration tool's **Create Replication Control Tables** function, *before* defining any source tables on the source server.

The control tables that the Capture triggers use on the source server are:

Register control table

Contains the name of the source table and the name of the CCD table that is associated with the source table. Also contains markers that track the sequence in which changes to the source table occur. There is one register table for each source server.

Pruning control table

Contains information about how far through the CCD table the Apply program has progressed in replicating changes to the target tables. One of the Capture triggers, the pruning trigger, uses this information to prune the CCD table. For more information on pruning, see "Pruning the CD and CCD Tables" on page 183.

The staging table that the Capture triggers use is called the CCD table. The CCD table contains the changed data that the Capture triggers write, and is a source table itself for update copies. There is one CCD table, located on the source server, for each source table.

The control and staging tables are described in more detail in the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*.

The Apply Program

The Apply program actually copies the changed data from the source table to the target table. You can run the Apply program at any server that has connectivity to the source and target servers, but the Apply program generally runs at the target server. See “Pull versus Push Apply Design” on page 188 for more information on locating the Apply program. The way the Apply program works with the replication administration tool and DataJoiner is almost identical to the way the Apply program works with DB2 systems.

The Apply program:

- Reads the changed data that was previously captured and stored in a change data (CD) table or CCD table
- Creates spill files (local to the Apply program) to hold the changed data that is to be replicated
- Copies the changes to target tables

When the source table is a DB2 table, the Apply program reads data directly from source table when copying the entire source table data for a full refresh to the target table. When the source table is in a non-IBM database, the Apply program accesses the source table through the DataJoiner nickname for that non-IBM source table. The Apply program must do an initial full refresh before any insert, update, or delete transaction can be applied to a defined replication source.

The Apply program connects to the various servers (the control server, the source server, and the target server) to perform its tasks. It can be located anywhere in the network, as long as it can make the connections required. It retrieves the changed data with a block fetch and propagates the changed data with row inserts, updates, and deletes. As a result of this design, to achieve the best possible performance, the Apply program should be local to the target tables. When replicating to a non-IBM target table, the Apply program’s performance is best when it is located with the DataJoiner database that contains the nicknames for that non-IBM target table.

The Apply program runs independently of the replication administration tool, but uses control information that the replication administration tool creates. The primary control information that the Apply program uses is stored in control tables at the control server. The control tables that are used by the Apply program are described in “Control Tables Used by the Apply Program” on page 174.

The Apply program applies the changed data to target tables that are either on local servers (where the Apply program runs) or remote servers, working through DataJoiner nicknames if either the source or target tables are in a non-IBM database.

Figure 20 shows the Apply program's relationship with the source tables, the source server control tables, the subscription definition control tables, and the target table.

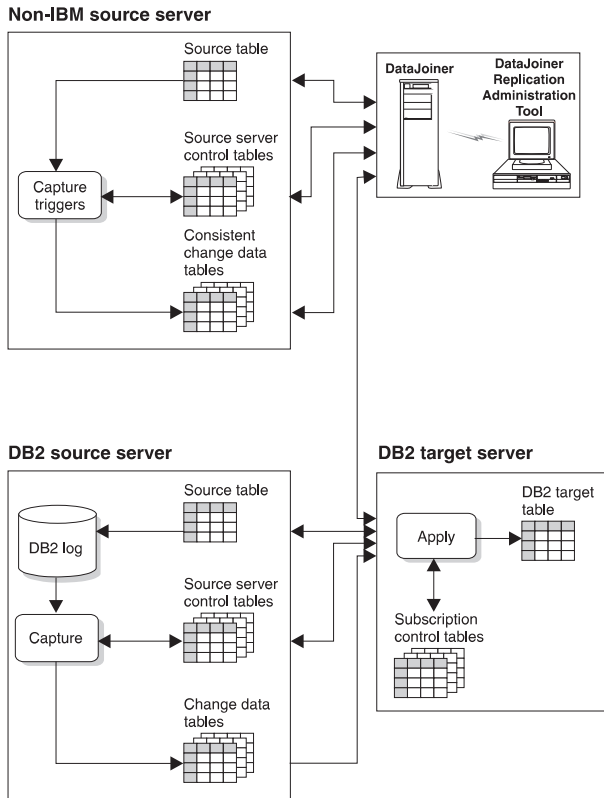


Figure 20. The Apply Program. The Apply program reads the data from the source server tables, manages the source server control tables, manages the subscription control tables, and updates the target table.

The Apply Qualifier

Several instances of the Apply program can share the same control server database. When started, the Apply program is supplied with an Apply qualifier, which is an arbitrary qualifier that is independent from a logon user ID. The Apply qualifier allows a logon user ID to run more than one instance of the Apply program. The Apply qualifier is one of the items that is specified in a subscription set.

Each subscription set has an associated Apply qualifier and one Apply qualifier can be associated with many subscription sets. As a result of this association, an Apply instance is responsible for processing only those subscription sets that have the matching Apply qualifier. You can have several instances of the Apply program running at the same time, each processing different subscription sets that are represented in the same control server. Using the Apply program password file feature⁶, you can have the Apply program use a different user ID and password when connecting to each of the servers, independent of the user ID in effect when starting the Apply process or job.

Control Tables Used by the Apply Program

The control tables that are used by the Apply program are located at the source server and the control server. When the source table is a non-IBM database, the Apply program accesses the control tables through DataJoiner nicknames.

The Apply program relies on the following control tables to manage the replication of source table changes to target tables:

Source server control tables

The control tables that are used by Capture triggers and the Apply program. These control tables (which include the register and pruning control tables⁷) must exist on any DB2 or DataJoiner database that is bound to the Apply program.

Control server control tables

The control tables that are located on the control server. They are:

Subscription set control table

Defines the characteristics of each subscription set, such as the subscription set name, the Apply qualifier that is associated with the subscription set, and the timing of the subscription set.

Subscription members control table

Contains information for an individual subscription member that maps a specific source table to a specific target table. Subscription sets of DB2 views usually have many subscription members for the same target table.

6. The Apply program password file is different from replication administration tool's password file.

7. The *register* table contains the name of the source table and its associated CD table. The *pruning control* table contains information about how much of the CD table the Apply program has replicated to the target tables; the Capture program uses this information to prune the CD table.

Subscription columns control table

Contains supplemental information about each copied column, such as renamed columns or computed columns.

Subscription events control table

Contains the timing information for processing the subscription set based on event triggering. A user application (or other subscription sets) maintains this table. This table itself can be replicated if you want to distribute event notifications. The replication administration tool supports continuous-timing processing of subscription sets.

Subscription statements control table

Contains the SQL statements or names of stored procedures to be run before or after the subscription set is processed.

Apply trail control table

Records statistics about refreshes and updates that are performed during each subscription set cycle.

Row-replica target list table

Maintains the names of the row-replica tables. This allows DB2 DPROP for Microsoft Jet to maintain a list of known row-replica tables in a DB2 or DB2 DataJoiner database. DB2 DPROP for Microsoft Jet uses this information during schema analysis to determine which, if any, row-replica tables should be deleted because the corresponding subscription member has been dropped since the last synchronization.

Subscription schema changes table

Used to signal modifications to a subscription.

Target Server Tables

The control tables that are located on the Microsoft Jet target server. They are:

Synchronization generations table

Used to prevent cyclic updates from propagating back to the RDBMS from a Microsoft Jet database.

Key string table

Maps Microsoft Jet table identifiers and row identifiers to primary key values.

Error messages table

Contains error codes and error messages.

Error information table

Contains additional information to identify the row-replica table and row that caused an error.

Error side information table

Contains the names of the conflict tables.

Conflict table

This table (one per target table, as needed at the target server) contains row data for DB2 DPROP for Microsoft Jet-detected conflict losers. If there is a conflict between the same row in the Microsoft Jet database (target server) and the source server, the row in the Microsoft Jet database "loses," is added to the conflict table, and is replaced by the row in the source.

These control tables are described in more detail in the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*.

The Apply Program and Control Servers

The control server is the logical server that contains the subscription control tables. Each Apply program is associated with a control server, which is specified during each call of the Apply program. Multiple Apply programs can share a control server.

Each control server can contain information about subscription sets that are associated with one or more Apply qualifiers. The qualifier associates a control server with an Apply program and one or more subscription sets.

The Apply Processing Cycle

Table 8 describes how the Apply program completes a replication cycle. In this example, the Apply program runs on the target server and connects to the control and source servers to access the control tables and to pick up the changed data.

Table 8. The Apply Program Cycle: A High-Level Overview of How a Subscription Set is Processed.

Step	Server
1. Check each subscription set associated with the Apply program's Apply qualifier to see if it is "eligible for processing", based on either time or an event.	Control server
2. For each "eligible" subscription set, check the sources for each subscription member to see if there is new, changed data to be copied to the target.	Source server
3. Pick up recent change data to be applied to the target tables for a subscription set.	Source server
4. Write the answer sets into a local "spill" file (possibly an in-memory file).	Target server

Table 8. The Apply Program Cycle: A High-Level Overview of How a Subscription Set is Processed. (continued)

Step	Server
5. Apply the change data in the spill file to the target tables.	Target server
6. Update subscription set status.	Control server
7. Report subscription member progress in the pruning control table.	Source server

Chapter 24. Data Replication Concepts

This chapter describes concepts specific to the replication administration tool. The concepts are similar to the advanced replication concepts that are described in the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*.

Subscription Members and Sets

A subscription member defines a source-to-target relationship between a source table and a target table. The member also defines the structure of the target table and specifies what is to be replicated to that target table. A subscription set contains the set's attributes and zero or more subscription members.

Subscription members that are associated with a particular source server *and* a particular target server can be grouped together in subscription sets. As a result of this grouping, the changed data for all the subscription members in a subscription set are replicated from their source tables to the specified target tables in one database unit-of-work.

One row in the subscription set control table identifies each subscription set. However, two rows identify a replica set. These rows differ from each other in one column that indicates the direction of replication.

See the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for more information about subscription rules and recommended uses.

Types of Copies

You can define subscription members with the replication administration tool to create the following types of copy tables, known as target table types:

User copy

A complete, condensed copy of the source table that must have a primary key.

Point-in-time copy

A complete, condensed copy of the source table at a certain point-in-time that must have a primary key. This table contains a time stamp column to indicate when a transaction occurred.

If the target is a non-IBM database, these two types (user copy and point-in-time) are the only two types supported.

When the target database is DB2, you can define two other types of copy tables: CCD (when a DB2 database is either source or target) and replica (when DB2 databases are both source and target). The replication administration tool does not support base aggregate or change aggregate types.

When the target database is Microsoft Jet, you define a row-replica table. Changes to this table are replicated back to the source table in an update-anywhere scenario. It differs from other replica tables in that conflicts are detected row by row, rather than by transaction.

See the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for descriptions of these other copy table types.

Full-Refresh and Differential Refresh

The Apply program copies data from the source to the target either by full-refresh copying or differential refresh copying.

During full-refresh copying, the Apply program:

1. Empties (deletes) all the rows from the target table
2. Reads all of the rows from the source table
3. Copies the rows to the target table

If the source table is in a non-IBM database, the Apply program accesses the source table through the DataJoiner nickname for the source table. You can specify full refresh only while defining a source table. If you have large tables, you might want to use a fastload program to simulate an initial full refresh copy. See the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for more information about fastload programs that can be used with the ASNLOAD Apply exit.

During differential-refresh copying, the Apply program copies only the changed data from the CD or CCD table to the target table. The first time the Apply program copies data to the target table, it does a full refresh to populate the target table. After the target table is populated, differential refresh is used.

Note: A non-complete CCD table is never fully refreshed because it contains only data that was changed in the source table.

Changed Data Tables

With the IBM Replication-Solution architecture, you can stage changed data; that is, place data in a table (called a staging table) to be accessed by another process (typically the Apply program). This section discusses two types of staging tables: CD tables and CCD tables.

When the source table is a DB2 table, the Capture program captures all the changes that are made to a source table and inserts change data rows into a CD table. The Apply program then pulls the committed changes from the CD tables, after which the Capture program automatically prunes change data rows from CD tables when they are no longer needed.

When the source table is in a non-IBM database, the Capture triggers capture committed changes that are made to a source table and insert change data rows into a CCD table. The Apply program then pulls the changes from the CCD tables. A pruning trigger (associated with the pruning control table) automatically deletes the change data rows from the CCD tables when the rows are no longer needed.

The CCD table that is created on a non-IBM source server, is built and filled differently than the CCD table discussed in *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*, but the Apply program uses both kinds of CCD tables in the same manner.

In this book, we refer to the CCD table that the DataJoiner's replication administration tool creates in non-IBM databases and maintains by the Capture triggers. The CCD table discussed in the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)* is maintained by an Apply process and is an optional, secondary staging table that is a copy of previously captured changes.

CD and CCD Tables

A CD table receives changed data rows from the Capture program when changes are made to a DB2 source table. Rows in a CD staging table reflect changes that are equivalent, if not identical, to the original operational updates. Both uncommitted and incomplete changes can appear in rows in a CD table. The CD table has no knowledge of transaction boundaries. Nor does the CD table know whether or not the transactions that issue the updates are committed, are incomplete, or are in flight.

The Apply program joins the CD table with the unit-of-work (UOW) control table to determine which changes in the CD table have been committed and are to be applied to the target table. Uncommitted changes that are in the CD table are pruned as the changes to the source table are aborted.

Rows in a CCD staging table reflect the committed results from the Capture triggers when an insert, update, or delete operation takes place against a non-IBM source table. CCD staging tables contain only committed change data. The Apply program queries the register control table to find out what committed changes have not been copied yet to the target table. After the Apply program has this information, it retrieves the appropriate committed changes from the CCD tables and applies them to the target table. Figure 21 shows the columns in a CCD table.

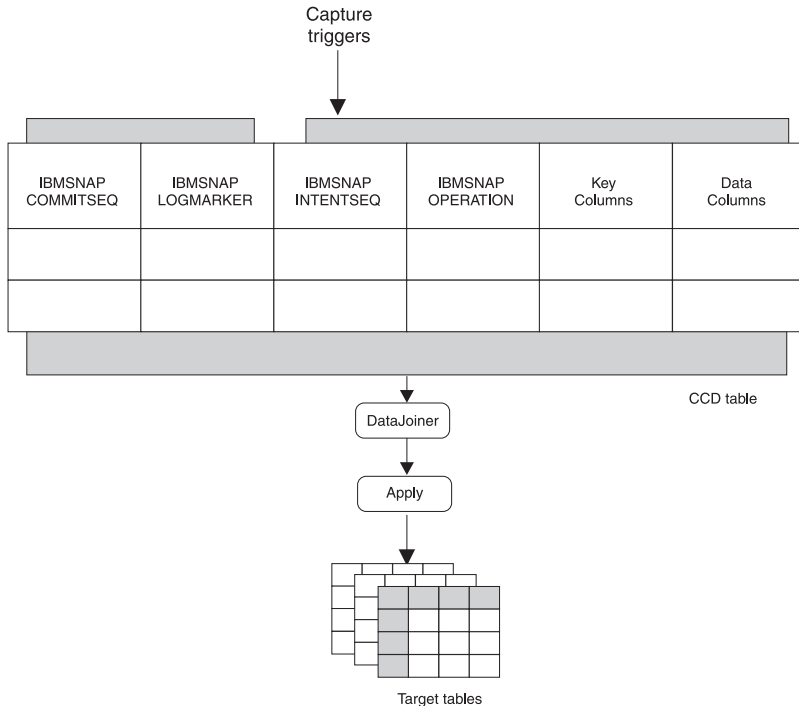


Figure 21. The CCD Table. The Apply program uses information in the CCD table to determine what to replicate.

If the source table is in a non-IBM database:

- The CCD staging table is local to the source table. If the source is a DB2 database, the CCD staging table can be local to, or remote from, the original source table. See the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for information on staging tables that are local or remote to the original source table.
- Two types of CCD tables are used. Generally, non-complete CCD staging tables are used when replicating changes to a target table. However, if the CCD staging table needs to be equivalent to a CD table that is maintained by the Capture program (in a DB2 environment), non-condensed CCD staging tables are used. Non-condensed CCD tables contain a row for each

insert, update, and delete to the source table with which the CD table is associated. See the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for a complete description of these types of staging tables.

- The non-complete CCD staging table (maintained by non-IBM triggers) is used only when replicating changes, not when the Apply program performs the full, initial refresh to a target table. When the Apply program performs a full, initial refresh to the target table, the Apply program uses the source table itself.

Pruning the CD and CCD Tables

When the source table is a DB2 table, the Capture Program prunes the CD table based on information inserted into the pruning control table by the Apply program. In this scenario, the Apply program maintains the CCD tables and *does not* automatically prune them. Refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for information about how these CCD tables are maintained and pruned.

When the source table is a non-IBM table, the Capture triggers prune the CCD table based on a synchpoint that the Apply program inserts into the SYNCHPOINT column of the pruning control table. Initially, when the Apply program performs a full refresh, it sets the synchpoint value to zero. A zero value signals the Capture triggers to start capturing changes to the source table. When the Apply program copies changes from the CCD table to the target table, the Apply program updates the SYNCHPOINT column. The pruning control trigger can then prune entries in the CCD table up through the row with the highest synchpoint value for changes that have already been replicated.

For the Capture program, you control whether or not pruning occurs by using the PRUNE or NOPRUNE parameter. You also control when the pruning takes place and how the prune interval is set in the tuning parameters table. See the Capture and Apply chapter for your platform in the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for more information about these parameters.

Transaction-Based versus Transaction-Consistent Replication: Using Internal CCD Tables to Reduce Network Load

The replication administration tool supports both *transaction-based replication* (replication of every update used by every transaction) and *transaction-consistent replication* (replication of only the committed results of the recent activity).

Figure 22 illustrates the difference between the two types.

```
Transaction 1: Update table1 set col1 = 'X' where key1 = 425
               Update table2 set col2 = 'B' where key2 = 425
Transaction 2: Update table1 set col1 = 'Y' where key1 = 425
Transaction 3: Update table1 set col1 = 'Z' where key1 = 425
```

Figure 22. Transaction-Based versus Transaction-Consistent Replication

In transaction-based replication, all four transactions are captured and replicated. In transaction-consistent replication, only the second update in Transaction 1 and Transaction 3 are replicated.

Transaction-consistent replication is superior to transaction-based replication because it produces the same change data results with fewer updates replicated. This type of replication reduces network load and can increase the availability of the target table.

You implement transaction-consistent replication by using the CCD tables created by the replication administration tool (the CCD tables maintained by the Apply program that are *local* to a CD table, not the CCD tables maintained by the Capture triggers). The CCD tables that are created by the replication administration tool are:

- Internal (local to the source table)
- Non-complete (containing only the rows involved in the replication process)
- Condensed (keeping only a copy of the latest captured value for each row)

The Apply program uses this CCD table as the source for replicating changes to the target table. The subscription sets are transaction consistent.

Before-Image Columns and After-Image Columns

A before-image column contains the value of a data column in a source table *before* that data column is updated. Similarly, an after-image column contains the value of a data column in a source table *after* the value in that data column is updated. Most database managers log both the before-image and after-image columns of the source table for each change to that table. The replication administration tool's Capture triggers can capture both before-image and after-image columns. Before-image columns can be useful to applications that require auditing or rollback capability.

See “IBM Replication Solution Restrictions” on page 193 for a list of the restrictions associated with before-image columns.

Before-Image Column Values

When you define a replication source and specify that you want to capture the before-image value of the source table, the before-image and after-image columns in the CD or CCD table have the following values when the source column changes:

Action	Column Value
Insert	The before-image column has a NULL value.
Update	Column values before the change are captured in the before-image columns; values after the change are in the after-image columns.
Delete	Both the before-image and after-image columns contain the before-image value.

NULL Values in Before-Image Columns

When the target table that will have before-image columns in it is initialized, before-image columns have NULL values. Full refresh copies are from the source table; therefore no before-image values are available. For example, the before-image columns of a point-in-time target table have NULL values if the source table has no updates because no before-image columns were copied.

Subsetting Target Tables

The replication administration tool supports both vertical and horizontal subsetting of the source table. Therefore, you can specify that only a subset of the source table columns and rows are to be replicated to a target table, rather than all of the columns and rows.

Vertical Subsetting

In some replication scenarios, you might not want to replicate all of the columns in the source table or view. This is appropriate, for example, if some of the columns in the source table are very large (like LOBs) or if the column data types are not supported by the intended target table. When the columns that actually get replicated are a subset of all the columns in the source table, this is called *vertical subsetting*.

See “IBM Replication Solution Restrictions” on page 193 for the restrictions associated with vertical subsetting.

You can implement vertical subsetting in either of the following two ways.

- While defining a source table, you can select only the columns that you want to be *available* for replication to a target table. Data for the unselected columns will not be copied into the CD or CCD table for this source table.

Note: CD or CCD tables must contain sufficient key data for point-in-time copies, which are maintained with key-qualified predicates.

- While defining a subscription member for a source table (which results in the creation of that subscription member), use the **Add a Member to Subscription Sets** option to select, from the available source table columns, which columns you want to actually replicate to the target table for that subscription member.

Horizontal Subsetting

You can also subset the rows from the source table that are to be copied to the target table by specifying a row predicate when defining the subscription member for the target table. This predicate is in the form of a “WHERE clause” and is used to select the rows from the source table that will be copied to the target table.

The replication administration tool assumes that target table primary keys will not change. When the primary key of a non-IBM source table is updated, the Capture triggers place an UPDATE row in the CCD table for the non-IBM source table. When the Apply program reads this UPDATE row, the Apply program first attempts to perform a searched UPDATE, which will likely fail with a row not found condition. The Apply program then automatically converts the failed UPDATE into an INSERT. The target table then can have both the original row with the original key *and* the updated row with the new key. The original key no longer exists in the source table.

When the primary key of a DB2 source table is updated, the Capture program places an UPDATE row in the CD table for the source table. If one or more of the columns that are defined in the predicate of a DB2 source table are updated, you must use the Capture program function that defines updates as DELETE and INSERT statements. To learn how to use this option, see “Replication Logical Partitioning Key Support” in Chapter 2 of *IBM DB2 Replication Guide and Reference (V5)* or “Enabling replication logical-partitioning-key support” in Chapter 6 of *IBM DB2 Replication Guide and Reference (V6)*. The replication administration tool’s Capture triggers do not support the logical partitioning key feature. Updates will not be treated as DELETE and INSERT.

Combining Data

One of the important features of the IBM replication solution is that you can create and maintain target tables with contents that are joins or unions of existing source tables.

The IBM replication solution enables you to create:

- Joins of tables from a single DB2 source server (using views)
- Unions of tables from one source server
- Unions of tables from multiple source servers, sometimes referred to as multisite unions

You can perform a join of rows from source tables in a single source server by defining a view as a source and within this view definition, performing multi-table joins. With views you can also combine data by:

- Altering source definitions or subscription members
- Performing computations on one or more existing columns in the source table as the column is replicated to the target table

You can create a logical union of two or more source tables all residing on the same source server. This is done by having multiple subscription members within the same subscription set. Each subscription member is a source of the union.

You can create a logical union of two or more source tables that reside on different source servers. This is a more complicated scenario.

The technique for creating this logical union requires you to:

- Create target tables and views

The target table must be partitioned through views such that each view isolates the rows that come from one of the source tables. Each target view must isolate its rows from the other target views.

The target view defines the target rows that will be copied from a particular source table.

- Define subscriptions

Each subscription member causes the changed source data to be replicated to a *target* view. Each subscription member lists the appropriate target view as the target table, not the target table itself.

Multiple subscription members are in multiple subscription sets because there are multiple source servers.

SQL Statements and Stored Procedures

You can specify SQL statements or stored procedure calls that are to be run before or after the Apply program processes a subscription set. This feature is useful for manipulating change data and controlling the sequence in which subscription sets are processed (signalling subscription events). There are three types of run-time processing statements:

- At the source server, before the changed data is fetched (opening of the answer set cursors)
- At the target server, before the changes are applied
- At the target server, after the changes are applied

If the source table is in a non-IBM database, DataJoiner processes the SQL statements. The stored procedures use the SQL CALL statement, supported by the replication administration tool, without parameters. The procedure name must be eight characters or less in length. The run-time procedures of each type are processed together in a single unit-of-work. Acceptable SQLSTATES can be defined for each processing statement.

Pull versus Push Apply Design

A pull versus push configuration is defined as where (physically) the Apply program is running, either at the source server or the target server. In the push method, the Apply program runs at a server other than the target server and pushes changed data to the targets. In the pull method, the Apply program runs at the target server and pulls changed data from the source. One Apply program could be pushing for some subscription sets and pulling for others.

When the Apply program processes a subscription member, it first connects to the source server in order to fetch the current changed data. If the source table is in a non-IBM database, the Apply program connects to a DataJoiner database (with DataJoiner connected to the non-IBM database) as the source server and accesses the data through DataJoiner nicknames. The changed data is fetched into a spill file that is local to the Apply program. When the data is retrieved, the Apply program connects to the target server and applies the changes, one row at a time, as an insert, update, or delete operation to each target table.

If the target table is in a non-IBM database, the Apply program connects to a DataJoiner database (with DataJoiner connected to the non-IBM database) and applies the data to the target table through DataJoiner nicknames.

Figure 23 on page 189 and Figure 24 on page 190 show push and pull scenarios. Notice that in both figures, the box that represents DataJoiner is labeled as the source server. This is from the Apply program's point of view. The Apply program accesses the target tables through DataJoiner; that is, to the Apply program, DataJoiner *is* the source server.

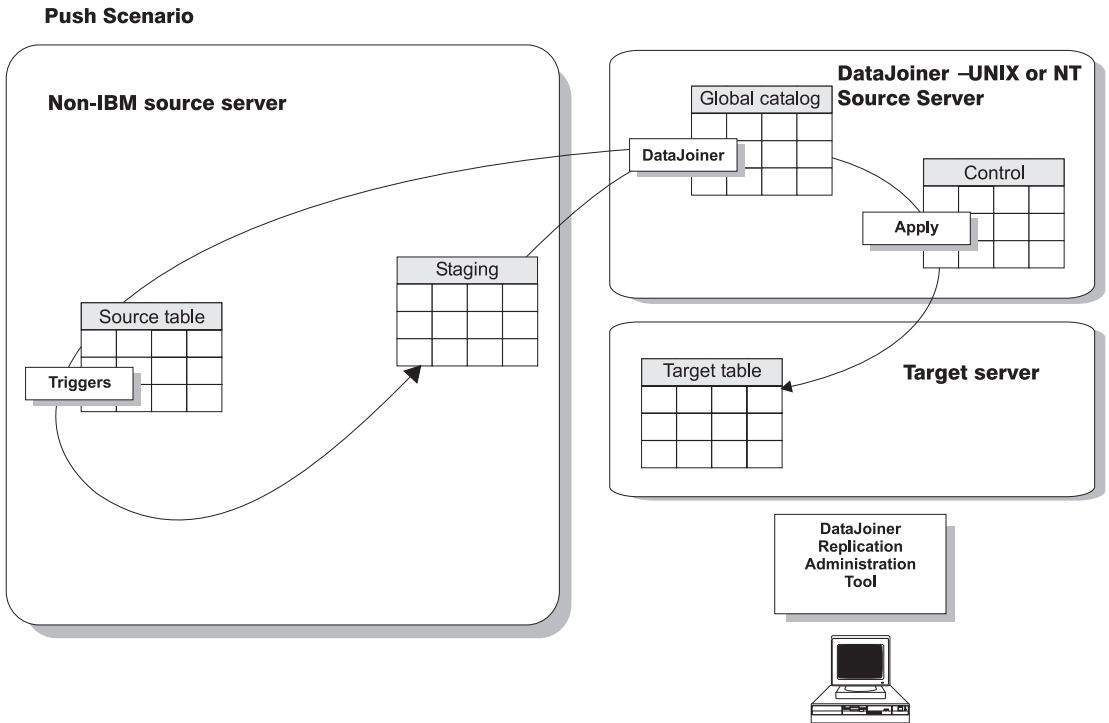


Figure 23. The Apply Program Push Scenario

Pull Scenario

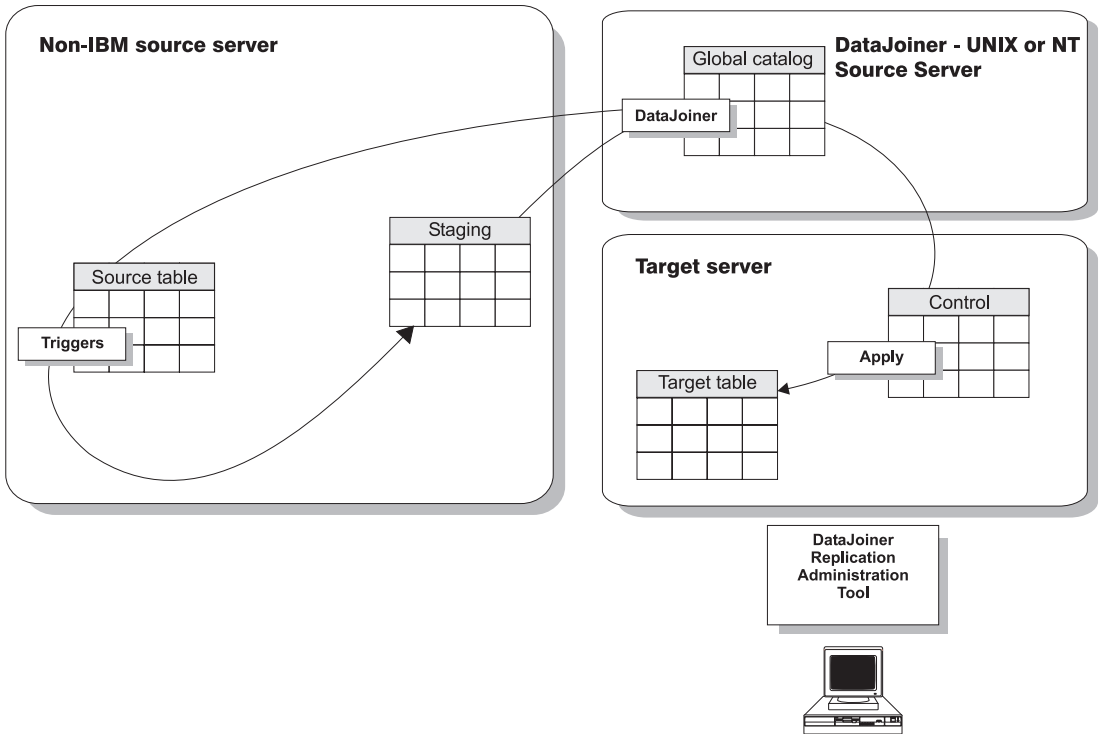


Figure 24. The Apply Program Pull Scenario

In pull mode, the Apply program connects to the remote source server (or to a DataJoiner database if the source table is in a non-IBM database) to retrieve the data. When all data is retrieved, the Apply program connects locally to the target server and applies the changes to the target table. The row-by-row process occurs as a local operation.

In push mode, the Apply program connects to the local source server, which is a DataJoiner database for non-IBM source servers. Then the Apply program connects to the remote target server (through DataJoiner if the source table is in a non-IBM database) and pushes the updates to the remote target table. The row-by-row process occurs as a remote operation.

The only thing you need to do to configure a push or pull configuration, is to decide where to run the Apply program. DataJoiner with the replication administration tool recognizes both configurations. DataJoiner with the replication administration tool automatically sets up the control tables so the Apply program can push or pull data.

Generally, a pull configuration performs better than a push configuration because the insert, update, and delete operations are performed to a local target rather than over a network. This is a more efficient use of the network. However, under the following circumstances a push configuration is a better choice:

- When there is no Apply program installed on the target server. In this instance, you might want to use the Apply program at the source server and *push* the data to the target server.
- The source table changes very infrequently, but when it changes it should be replicated as soon as possible.

Update-Anywhere Replication

The replication administration tool supports update-anywhere replication only for DB2 sources and targets. Non-IBM sources or targets are not supported. Refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for information on update-anywhere replication within the DB2 database family.

Views for Sources and Targets

You can define source views only for DB2 sources. You can define target subset views only for DB2 targets.

Source Views

Join views fill many requirements: denormalizing (restructuring) copies in data warehouse scenarios (thus enabling easier querying of copied data), and addressing the routing problem, sometimes called the database partitioning problem in distributed computing scenarios. For example, knowing where to send a bank account update may require a join of the account table with the customer table in order to know which branch of the bank the customer deals with. Typically, production databases are normalized so that the geographic details, such as branch-number, are not stored redundantly throughout the production database.

Using DataJoiner, you can create a join. Using the replication administration tool, you can define a join as a source table. The joins can only include tables that are defined as source tables. If the source tables defined in the join have CCD tables, a CCD view is created from the source server control tables.

DataJoiner with the replication administration tool supports the following types of view definitions:

- Simple inner-joins over one or more defined source tables

- Simple inner-joins over CCD staging tables that are defined as source tables and maintained by an Apply program or an application other than the replication administration tool and an external data source, such as DataPropagator NonRelational and IMS source data
- Only DB2 views or views of tables that reside within DataJoiner databases. Views of tables that are stored on Oracle, Microsoft SQL Server, Sybase, or Informix are not supported.

Target Views

You use the target views only when consolidating data from multiple sources. Multiple source tables are updated and consolidated into a union at the target server. The replication administration tool supports these types of views, even though it does not create them.

Preventing Gaps in the Target Table

The Capture triggers operate synchronously; therefore you do not experience gaps between the capturing of the changed data and the replication of that data to the target table. However, under certain circumstances during DB2 replication, gaps can occur if you stop running the Capture program. Refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for information about these circumstances.

Coordination of Source Security and Target Security

In a heterogeneous environment, each system has its own security system. These security systems need to be coordinated so the replication data can flow from the source database to the target database. DataJoiner coordinates the security according to how the security information from each system is defined to DataJoiner. See “Mapping of User IDs” on page 206 for more information about the security necessary for data replication and how the replication administration tool and DataJoiner enable this security.

Promote Functions

You can use the promote functions from the replication administration tool to reverse engineer your tables, registrations, or subscriptions, in order to create a script file with appropriate data definition language (DDL) and data manipulation language (DML). These functions are useful when you define tables, registrations, or subscriptions on one system (a test system, for

example), and you need to copy the replication environment to another system (a production system, for example). Table 9 shows the three promote functions.

For example, use the promote functions to define subscriptions for remote DB2 Personal Edition targets. After you define a model target system in your test environment, you can create subscription scripts (and modify the Apply Qualifier and so on) for your DB2 Personal Edition systems, which are not otherwise supported from a central control point.

Table 9. Promote Functions

Promote table	<p>This function promotes tables, table spaces, and indexes. It does not promote constraints defined for a table.</p> <p>This function is fully supported for UDB V5, but for the IBM Common Server you can only promote tables, not table spaces.</p>
Promote registration	<p>This function promotes registrations and view registrations from a source server.</p>
Promote subscription	<p>This function promotes subscriptions, subscription sets, subscription members, subscription columns, subscription prune control, and subscription statements.</p> <p>From the Promote Subscriptions window, you can change your subscriptions (before promoting them) by setting new values for any of the following fields:</p> <ul style="list-style-type: none"> Apply Qualifier Set Name Source server Source alias Target server Target alias Control server Control alias

IBM Replication Solution Restrictions

Keep the following restrictions in mind when you are planning to use the replication administration tool in your replication scenarios:

- Check the restrictions of all the products in your replication environment to make sure that your use of the replication administration tool is consistent with those restrictions.
- You must have one DataJoiner database for *each* non-IBM source server.

Although one DataJoiner database can be used as a source table for replicating data to more than one non-IBM target server, a unique DataJoiner database is needed for *each* non-IBM source server. The reason for this is as follows.

For every replication scenario, there is a set of control tables, each with names that cannot be changed. When replicating to non-IBM target servers, none of these control tables needs to be located in the non-IBM database because a DataJoiner database is the target for the Apply program. The nicknames used here refer to the target table and not to any of the control tables.

For non-IBM source servers, however, some of the control tables *must* be located in the non-IBM database so the Capture triggers can update them. Because of this location requirement, the DataJoiner nicknames associated with those control tables must be the actual control table names, and their schema must be ASN. Because a DataJoiner database cannot contain more than two identical nicknames with identical schemas, one DataJoiner database must be used for each non-IBM source server. You can, however, support multiple non-IBM source servers within one DataJoiner instance by creating multiple DataJoiner databases within that one DataJoiner instance.

- The control tables that are used by Capture triggers and the Apply program must exist on any DB2 or DataJoiner database to which the Apply program is bound.
- You can define source views only for DB2 sources. You can define target subset views only for DB2 targets.
- The replication administration tool does not support:
 - Double-byte character character set (DBCS)
 - Synchronous replication (replication of data within the same unit of work as the original change to the source database). If synchronous data replication is essential to your application, use DRDA two-phase commit within the application.

However, the replication administration tool works asynchronously and therefore does not affect the application that is making the original change to the source database. By working asynchronously, the replication administration tool provides you with the benefits of staging the changed data (for example, better network use, less database contention, and the opportunity to enhance the data as it is replicated).

- Delimited names.
- Update-anywhere replication for non-IBM sources and targets.
- Microsoft SQL Server's ability to access a database by defining an alias for the user.
- The Capture triggers do not support optional processing updates as delete and insert statements.
- The restrictions associated with before-image columns are:

- Before-image columns in a target table cannot be designated as primary key columns of the target tables.
- Before-image columns cannot be referred to in subscription member predicates for user copies and point-in-time copies.
- Before-image columns cannot be used in replicas or row replicas

Chapter 25. Replication Requirements

The following sections describe the replication administration tool hardware and software requirements.

Supported Sources and Targets

You can use the replication administration tool to set up replication between the databases that are listed in Table 10:

Table 10. Supported Sources and Targets

Database	Capture and Apply Programs needed
DB2 UDB V5 or V6	Includes Capture and Apply programs. The Capture program is required when DB2 UDB is used as source. The Apply program is required when the Apply program is located at the DB2 UDB target.
DB2 for MVS V3R1, DB2 for MVS V4R1, and DB2 for OS/390 V5R1 or V6	The Capture program for MVS is required when DB2 for MVS or DB2 for OS/390 is the source. The Apply program for MVS is required when the Apply program is located at the DB2 for MVS or DB2 for OS/390 target.
DB2 for common servers V2 and DataJoiner V2	The Capture program is required when DB2 for common servers or DataJoiner is the source. The Apply program is required when the Apply program is located at the DB2 for common servers or DataJoiner target.
Oracle V7.0.13 or later (UNIX) and Oracle V7.3 or later (Windows NT)	The replication administration tool creates Capture triggers on an Oracle source. The Apply program is required when Oracle is the target (with DataJoiner as the target server).
Informix V7.1 or later (UNIX) and Informix V7.2 or later (Windows NT)	The replication administration tool creates Capture triggers on an Informix source. The Apply program is required when Informix is the target (with DataJoiner as the target server).
Sybase V4.6 or later (UNIX) and Sybase V11 or later (Windows NT)	The replication administration tool creates Capture triggers on an Sybase source. The Apply program is required when Sybase is the target (with DataJoiner as the target server).
Sybase SQL Anywhere 5.0 (Windows NT)	The replication administration tool creates Capture triggers on an SQL Anywhere source. The Apply program is required when SQL Anywhere is the target (with DataJoiner as the target server).

Table 10. Supported Sources and Targets (continued)

Database	Capture and Apply Programs needed
Microsoft SQL Server V4.21 or later (UNIX) and Microsoft SQL Server V6.0 or later (Windows NT)	The replication administration tool creates Capture triggers on an Microsoft SQL Server source. The Apply program is required when Microsoft SQL Server is the target (with DataJoiner as the target server).
Microsoft Jet	DB2 DPROP for Microsoft Jet includes Capture, Apply, and Administration functions in a single program.

DataJoiner support for data sources varies by platform. DataJoiner also has client software requirements for each data source. See “Requirements for DataJoiner Data Sources” on page 26 for a list of clients supported by DataJoiner.

The Replication Administration Tool Hardware Requirements

The hardware requirements are:

- A personal computer that supports Windows NT or Windows 95 or Windows 98
- 24 MB RAM (32 or more is recommended)
- 2 MB hard drive storage

DB2 DPROP for Microsoft Jet Hardware Requirements

DB2 DPROP for Microsoft Jet runs on any IBM PC-compatible hardware with a minimum of 16 MB of storage. The program requires 1 MB of disk space. In addition, DB2 DPROP for Microsoft Jet requires additional storage space for:

- Microsoft Jet tables
- Spill files for fetching answer sets
- Log and trace files

You install DB2 DPROP for Microsoft Jet on the client as part of the DataJoiner Replication Administration tool installation process.

The Replication Administration Tool Software Requirements

You can set up your the replication administration tool environment in a variety of ways. This section shows the most common ways to set up your environment and the software requirements that are associated with each environment setup. Figure 25 on page 199 shows software requirements if DataJoiner for Windows NT and the replication administration tool are

located on the same workstation. Figure 26 on page 200 and Figure 27 on page 201 show software requirements if DataJoiner and the replication administration tool are on separate systems.

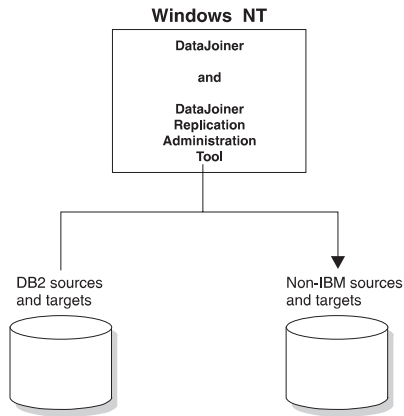


Figure 25. DataJoiner and the Replication Administration Tool on a Single Windows NT Workstation.

In Figure 25, DataJoiner provides access to DB2 databases. DataJoiner contains the same function as CAE V2 and DDCS V2. With this configuration, the replication administration tool software requirements are:

- Microsoft Windows NT V3.5.1 or later with Microsoft service pack 3
- DataJoiner for NT V2

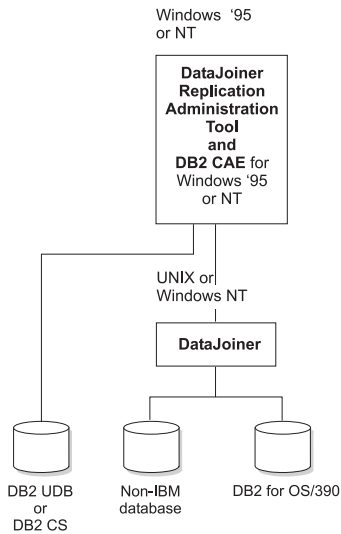


Figure 26. DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses DataJoiner to Connect.

In Figure 26, the replication administration tool installed on Windows NT, 95, or 98 works with DataJoiner for UNIX or Windows NT to access DB2 and non-IBM sources and targets. With this configuration, the replication administration tool software requirements are:

- Microsoft Windows 95 with Microsoft service pack 2, Windows 98, or Windows NT V3.5.1 or later with Microsoft service pack 3
- Client Application Enablers (CAE) V2.11 or later for Windows NT, Windows 95, or Windows 98

DataJoiner V2 for UNIX or Windows NT provides access to DB2 for OS/390 and non-IBM sources and targets. If you have DB2 UDB or DB2 for common servers on the same workstation that you have the replication administration tool, you do not need to install CAE. It is provided with DB2.

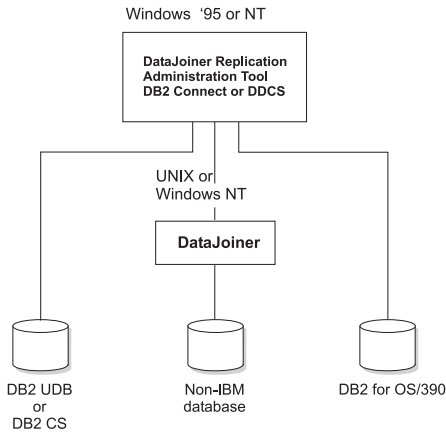


Figure 27. DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses the Replication Administration Tool to Connect.

In Figure 27, the replication administration tool provides access to DB2 for OS/390. With this configuration, the replication administration tool software requirements are:

- Microsoft Windows 95, 98, or Windows NT V3.5.1 or later
- DB2 Connect or DDCS for common servers V2R3 or later, to access DB2 for OS/390 sources and targets

DataJoiner V2 for UNIX or Windows NT provides access to non-IBM sources and targets. CAE is provided with DB2 Connect. DDCS prerequisites DB2, and DB2 provides CAE.

DB2 DPROF for Microsoft Jet Software Requirements

To run DB2 DPROF for Microsoft Jet, the following software must be installed on the client machine:

- Microsoft Windows NT, Windows 95, or Windows 98
- Microsoft Jet or Microsoft Access for Windows NT (or for Windows 95 or 98)
- DB2 Client Application Enabler (DB2 CAE)
- DB2 DPROF for Microsoft Jet

For information about installing Microsoft products, refer to the documentation that is shipped with that product. For information about installing DB2 Client Application Enabler, see the documentation available with the product.

To run DB2 DPROP for Microsoft Jet, the DataJoiner Replication Administration tool must be installed on the control server.

Capture and Apply Program Requirements

When you replicate from a non-IBM source, the replication administration tool provides Capture triggers. When you replicate from a DB2 source, the Capture program is provided as part of the DB2 package or is separately orderable. The Apply program for UNIX is provided with DataJoiner for UNIX. The Apply program for NT is provided with DataJoiner for NT. When you replicate to a DB2 target, the Apply program is provided as part of the DB2 package or is separately orderable.

Requirements for the Capture and Apply programs are described in the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V5)* and the *IBM DB2 Replication Guide and Reference (V6)*. Table 11 lists the Capture and Apply programs supported by DataJoiner with the replication administration tool.

Table 11. Summary of Capture and Apply Programs Supported by DataJoiner with the Replication Administration Tool

The Capture Program	The Apply Program
Capture for UNIX	Apply for UNIX (you can select Apply during DataJoiner custom installation)
Capture for Windows NT	Apply for Windows NT (Apply for Windows NT is included with DataJoiner)
The V5 Capture program	The V5 Apply program
Capture for OS/2	Apply for OS/2

Authorization Requirements

The replication administration tool uses mechanisms already in place for database security. This section describes the authorizations that the replication administration tool and DataJoiner use for the:

- Administration of replication
- Capture program
- Apply program

A sample set of privilege levels for replication user IDs is provided in “System-Specific Privilege Levels” on page 205.

Authorization Requirements for Administration

For each database involved in replication, the administrator who is working with replication must have sufficient privileges to define replication sources and targets. As administrator, you must be able to:

- Select catalog tables of source or target databases.
- Create control tables at the source, target, or control servers.

For non-IBM databases, you must be able to:

- Create tables.
- Create Capture triggers on:
 - Source tables
 - Control tables
- Create procedures.
- Create sequences.⁸
- Select system tables.
- Select catalog tables.

For DB2 databases, you must be able to:

- Create tables.
- Create table spaces.
- Bind plans on each DB2 database involved in replication, including the source server, target server, and control server.

Most users set the replication administration tool to use the ID and password of a database administrator to connect to source, target, and control servers. The section “Setting Administrative Preferences” on page 219 describes how to set up one or more IDs and passwords in the replication administration tool. To complete the set up of your replication environment you need to map your authorizations through DataJoiner. See “Mapping of User IDs” on page 206.

Authorization Requirements for the Capture Program

For heterogeneous replication, the replication administration tool creates Capture triggers to capture changes from the source. Capture triggers do not require additional authorization. If your source is on a DB2 database, you use the Capture program to capture changes; some authorization is required. Refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for Capture program authorization and security issues.

8. In Oracle only.

To complete the set up of your replication environment you need to map your authorizations through DataJoiner. See “Mapping of User IDs” on page 206.

Authorization Requirements for the Apply Program

The user ID that will run the Apply program (Apply user ID) must be a valid logon ID on the source, control, and target servers. This user ID must be able to:

- Access the replication source tables.

If the source table is at a non-IBM database, the Apply user ID:

- Must be able to access the DataJoiner nickname associated with the source table
- Must be mapped to a user ID (through DataJoiner) that has access privileges for the source table at the non-IBM database

- Access the replication target tables.

The Apply user ID must have update privileges for the target table. For a non-IBM database, the Apply user ID:

- Must be able to update the DataJoiner nickname associated with the target
- Must be mapped to a user ID (through DataJoiner) that has update privileges for the target table at the non-IBM database

- Access and update all control tables that are generated by the replication administration tool and built at the source and target database.

For non-IBM databases, the register and pruning control tables have nickname equivalents in DataJoiner. The Apply user ID that is pushing or pulling changed data to a target:

- Must have access and update privileges to the nicknamed source control tables in DataJoiner
- Must be mapped to a user ID (through DataJoiner) that has both access and update privileges for the control table at the non-IBM database

- Have run privileges on the Apply program plan at the source, target, and control server.

With the proper authorization, any user ID can run any Apply program instance. The restriction of one user ID per Apply program instance has been removed.

Additional notes about the Apply program authorization:

- You can use different user IDs at your source, target, and control servers.
- On DB2 for MVS, the Apply for MVS load library must be APF-authorized.

- An Apply program running on DB2 Universal Database might require a password file to connect to the source or target server. See “Providing End-User Authentication in NT” on page 224 for more information.

To complete the set up of your replication environment you need to map your authorizations through DataJoiner. See “Mapping of User IDs” on page 206.

System-Specific Privilege Levels

This sections provides a sample set of privilege levels you can use for IDs used for replication.

DB2 UDB

You must have DBADM, CONTROL, or SELECT privileges at the source, target, and control server.

DB2 for common servers

We recommend SYSADM or DBADM privileges at the source, target, and control server.

DB2 for MVS or DB2 for OS/390

At the source, target, and control server, you must have CREATETS privilege and any of the following privileges: SYSADM, DBADM, or CREATETAB.

Oracle IDs are set up with a default table space (with unlimited quota), default temp table space of TEMP, and default profile of DEFAULT. They are granted a system privileges of CREATE SEQUENCE/SESSION/TABLE/TRIGGER/VIEW. No password is required to enable the role, and the operating system requires no authentication.

Sybase on UNIX

IDs are set up with a default database. They are part of a group that has access grants of CREATE DEFAULT/PROCEDURE/RULE/TABLE/VIEW.

Sybase on NT

IDs are set up with a default database. They are part of a group that has object permission filters of PROCEDURE/TABLE/VIEW and command permissions of CREATE DEFAULT/PROCEDURE/RULE/TABLE/VIEW.

Sybase SQL Anywhere

IDs are set up with a default database. They are part of a group that has object permissions filters of TABLES/VIEWS/STORED PROCEDURES on ALL OBJECTS.

Microsoft SQL Server

IDs are set up with a default database. They are part of a group that has object permissions filters of TABLES/VIEWS/STORED PROCEDURES on ALL OBJECTS.

Informix

IDs are set up with default database and database privileges of RESOURCE which gives them CREATE INDEX/PROCEDURE/SYNONYM/TABLE/VIEW privileges.

Mapping of User IDs

This section explains how user IDs used for replication activities are mapped, through DataJoiner, to source, target and control servers. For a high-level discussion of DataJoiner's mapping coordination, see "Coordination of Source Security and Target Security" on page 192. This section discusses DataJoiner's mapping of IDs on the level of replication administration tool preference connections, DataJoiner catalog tables, and Apply program password files.

When accessing a DB2 database, the replication administration tool or the Apply program must map their user ID and password to a user ID and password in the DB2 database. When accessing a non-IBM database, the replication administration tool and the Apply program must map their user ID and password to DataJoiner. And DataJoiner must then map the user ID and password, through user mapping, to non-IBM database.⁹

The replication administration tool user ID and Apply user ID must have sufficient authority:

- In DataJoiner, to work with replication control tables and nicknames.
- In non-IBM databases, to work with source, target, and control tables.

The ID that is used in DataJoiner does not have to be the same ID that is used in the non-IBM database because DataJoiner can translate IDs and passwords.

Figure 28 on page 207 shows what happens when the replication administration tool sets up replication to allow data to replicate from a Sybase source to a DB2 for OS/390 target.

9. If you are using Microsoft SQL Server in your replication environment, do not use alias user IDs. Alias user IDs will not map correctly.

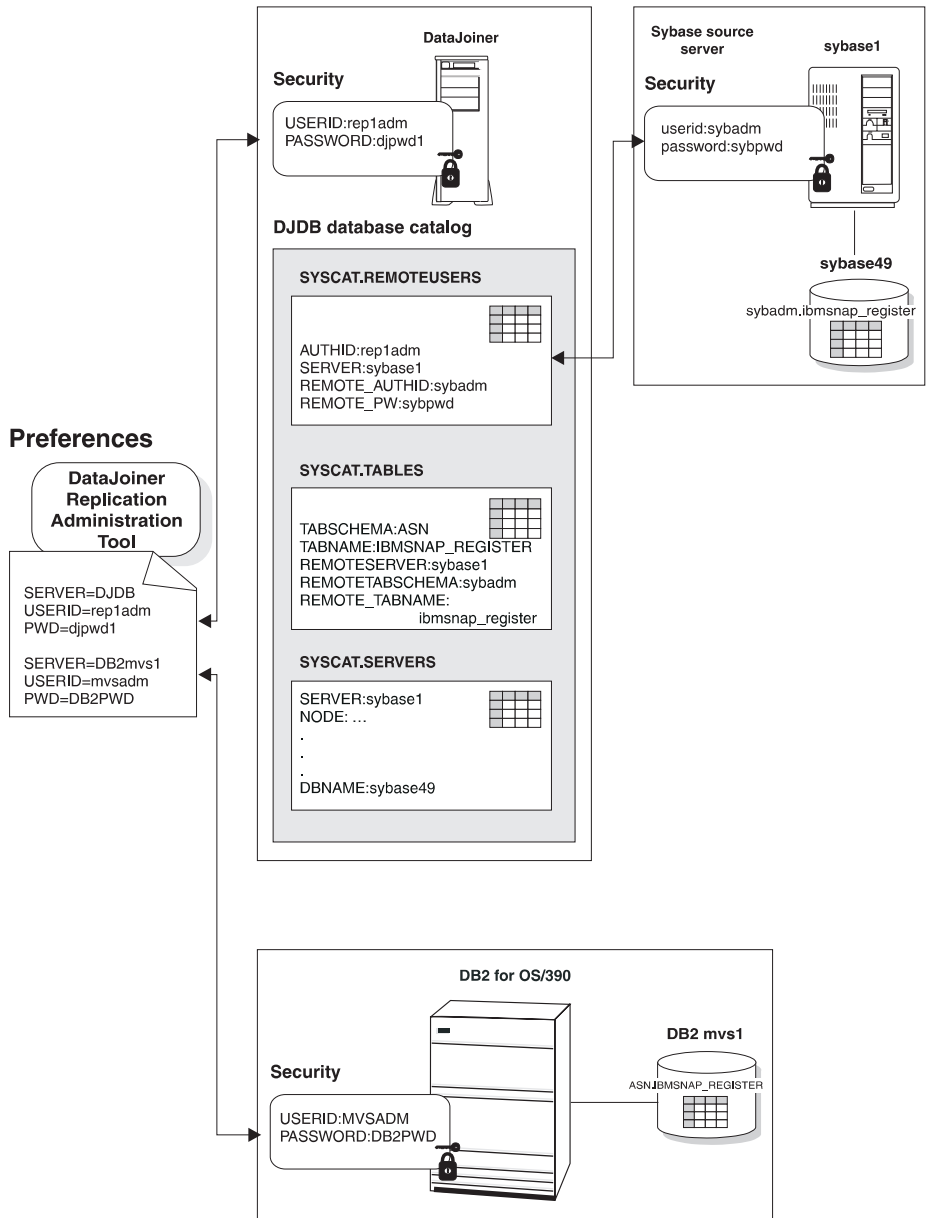


Figure 28. DataJoiner Maps the Replication Administration Tool User IDs to Source and Target. The user ID specified in the replication administration tool is mapped in DataJoiner to a non-IBM source and a DB2 for OS/390 target

In Figure 28, the administrator creates control tables and defines replication sources and targets using the replication administration tool. Mapping occurs between the user ID and password set in the replication administration tool's

Preferences folder and DataJoiner's SYSCAT.REMOTEUSERS catalog table. DataJoiner then connects to the Sybase database. If the user ID and password are valid, the control tables are placed at the Sybase database. To place control tables at DB2 for MVS, the replication administration tool connects directly to DB2, using the user ID and password set in the Preferences folder.

Figure 29 on page 209 shows what happens when the Apply program performs replication.

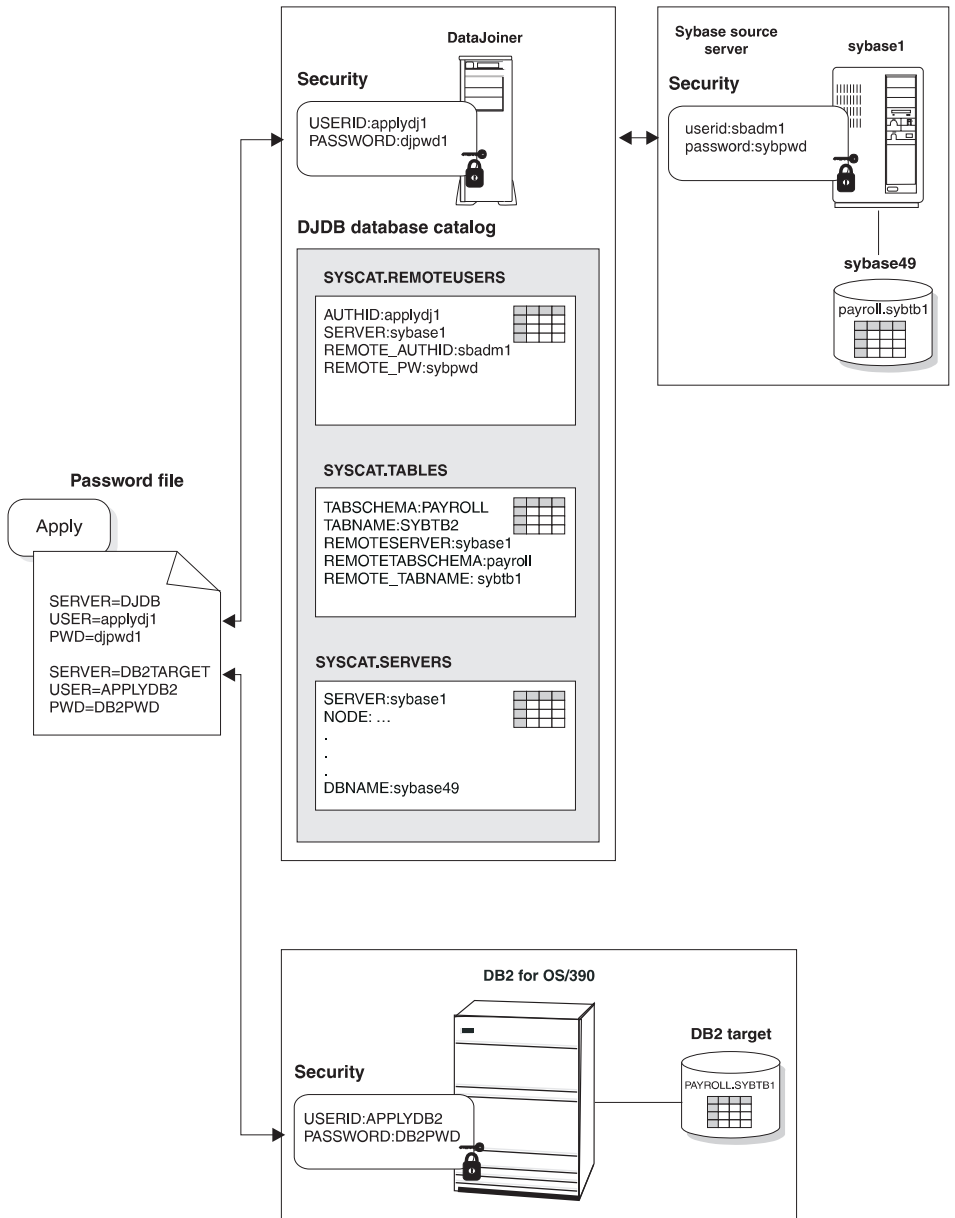


Figure 29. DataJoiner Maps User ID and Apply Password File to Connect to Source and Target. The Apply program passes user ID and password to DataJoiner, which uses them to access a non-IBM source and a DB2 for OS/390 target.

In Figure 29, the Apply password file specifies the database server, user ID and password. The Apply program connects to the DataJoiner database using the user ID and password from the Apply password file. DataJoiner then

searches its SYSCAT.REMOTEUSERS catalog table to find the Sybase user ID and password that is mapped to the connected user. DataJoiner then connects to the Sybase source database that uses the Sybase user ID and password. When replicating changes to DB2 for OS/390, the Apply program connects directly to DB2 for OS/390. The Apply program does not need to connect to a DataJoiner database to access DB2 for OS/390. The Apply program connects through DataJoiner DDCS or DB2 Connect. The entry in the Apply password file must be valid for DB2 for OS/390.

Chapter 26. Setting up the DataJoiner Environment for Replication

This chapter explains how to begin to set up your replication environment while installing and configuring DataJoiner. In this chapter, you:

1. Install DataJoiner.
2. Create a DataJoiner instance and, within the instance, a database for each non-IBM database that is a source for replication. Your target databases can use an existing database within the same DataJoiner instance, or you can create a new database for your targets.
3. Configure access to your data sources (your replication source, target, and control servers).
4. Connect clients to DataJoiner. If the Apply program that you are using for replication is remote from DataJoiner, you must connect the Apply program as a client to DataJoiner.

This chapter explains these steps for DataJoiner for Windows NT. After this chapter, you must proceed to “Chapter 27. Installing the Replication Administration Tool and Connecting All Databases” on page 217 to install the replication administration tool and connect all databases that are involved in replication.

Setting up DataJoiner for Windows NT

This section describes how to prepare DataJoiner for Windows NT for replication.

Installing DataJoiner

In “Chapter 4. Installing DataJoiner” on page 31, you have the option of installing the replication administration tool after you install DataJoiner. You do not have to select the Apply program. Apply for NT Apply for Windows NT is automatically installed when you install DataJoiner.

Setting up an Instance

In “Chapter 6. Getting Started with DataJoiner” on page 41, read sections “Working with DataJoiner Instances” on page 41 and “Starting and Stopping DataJoiner” on page 42. The DataJoiner databases that you set up for replication, reside on one DataJoiner instance. You must create one DataJoiner database for *each* non-IBM replication *source* server. You can support many

non-IBM replication *target* servers with one DataJoiner database. The DataJoiner databases that you set up reside on one DataJoiner instance. Use the DB2 **CREATE DATABASE** command to create each database.

In the section “Starting and Stopping DataJoiner” on page 42, you can configure Windows NT to automatically start your DataJoiner instance and DB2 security service:

- Select **My Computer** from the desktop
- Select **Services**
- Select the DB2 instance to be used for replication and the DB2 security server
- Select **Startup**

The information that is presented in the section “Running Your Own Applications” on page 47, applies to Apply programs installed on DataJoiner for Windows NT. For Apply programs installed on other platforms see “Connecting Clients to DataJoiner” on page 213.

Configuring Database Connections

Within “Chapter 7. Configuring Access to Data Sources” on page 51, several sections are applicable to replication. The Apply program in DataJoiner uses the connections you configure in this section to access source, target, and control servers.

Table 12. Configuring Database Connections in DataJoiner: NT

If Accessing	You need to	Refer to this section
DB2 UDB V5 or V6, or DB2 CS V2	Catalog DB2 in DataJoiner.	“Chapter 9. Accessing DB2RA Data Sources” on page 67.
DB2 for MVS or DB2 for OS/390	Catalog the DB2 database.	“Accessing DRDA Data Sources Using DataJoiner DDCS” on page 62.
Informix	Create a server mapping, user mapping, and nickname for Informix, within DataJoiner.	“Chapter 10. Accessing Informix Data Sources” on page 81.
Sybase	Create a server mapping, user mapping, and nickname for Sybase, within DataJoiner.	“Chapter 14. Accessing Sybase Data Sources Using Sybase Open Client” on page 97.

Table 12. Configuring Database Connections in DataJoiner: NT (continued)

If Accessing	You need to	Refer to this section
Sybase SQL Anywhere	Create a server mapping, user mapping, and nickname for SQL Anywhere, within DataJoiner.	“Chapter 15. Accessing Sybase SQL Anywhere Data Sources Using ODBC” on page 103.
MS SQL Server	Create a server mapping, user mapping, and nickname for Microsoft SQL Server, within DataJoiner.	Refer to the section, “Chapter 11. Accessing Microsoft SQL Server Data Sources Using ODBC” on page 85, to create a server mapping, user mapping, and nickname for Microsoft SQL Server, within DataJoiner.
Oracle	Create a server mapping, user mapping, and nickname for Oracle, within DataJoiner.	Refer to the section, “Chapter 13. Accessing Oracle Data Sources Using SQL*Net or Net8” on page 91, to create a server mapping, user mapping and nickname for Oracle, within DataJoiner.

You must define server and user mappings, as directed in “Chapter 7. Configuring Access to Data Sources” on page 51, for each DataJoiner database that requires access to a source or target. You must have one DataJoiner database for *each* non-IBM replication *source* server. You can support many non-IBM replication *target* servers with one DataJoiner database. The DataJoiner databases that you set up reside on one DataJoiner instance.

Connecting Clients to DataJoiner

“Client Access to DataJoiner” on page 27 applies to Apply programs installed on DB2 for MVS, DB2 UDB, or DB2 CS that must access non-IBM sources or targets through DataJoiner. Refer to the chapter applicable to your network protocol: IPX/SPX, NetBIOS, or TCP/IP. Later, in “Configuring Access from the Replication Administration Tool to DataJoiner and DB2” on page 219, you will connect the replication administration tool as a client to DataJoiner.

Client software must be installed and configured for each DataJoiner database that is created to replicate with a remote Apply program.

What to Do Next

Before proceeding to “Chapter 27. Installing the Replication Administration Tool and Connecting All Databases” on page 217, make sure that you install all Capture and Apply programs on the systems of your choice. Refer to

“Capture and Apply Program Requirements” on page 202 for a list that is supported by DataJoiner with the replication administration tool. And refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)* for instructions on installing the Capture and Apply programs.

Setting up DB2 DPROP for Microsoft Jet

To prepare the replication environment, you need to prepare both the server and the client environments. The following sections provide instructions for preparing the server and the client and for providing end-user authentication at the servers.

Preparing the Server

You prepare the server to use DB2 DPROP for Microsoft Jet just as you would for DataPropagator Relational. To prepare the server:

1. Create the necessary replication control tables by using the DataJoiner Replication Administration tool. See *DB2 Replication Guide and Reference*.
2. Define replication sources by registering the source tables on each source server by using the DataJoiner Replication Administration tool using the DataJoiner Replication Administration tool.
3. Define replication targets (subscriptions) on the control server by using the DataJoiner Replication Administration tool.
4. Start the Capture program for each DB2 source server, if applicable. For information about starting the Capture program, see *DB2 Replication Guide and Reference*.

Preparing the Client

To prepare the client, install the following software (if it is not already installed):

1. Install DB2 Client Application Enabler (CAE) and configure DB2 connectivity to the source and control servers for the appropriate communication protocols.
2. Configure the DB2 ODBC driver by using the DB2 Client Configuration Assistant window.
3. Install either one of the following:
 - Microsoft Data Access Components (MDAC) (downloadable from <http://www.microsoft.com/data/mdac15.htm>)
 - Microsoft Access

4. Install the DAO component (downloadable from <http://www.nesbitt.com/bctech.html> or available on the Microsoft Visual C++ Version 5 CD-ROM).
5. Install DB2 DPROP for Microsoft Jet (during DataJoiner installation).
 - During installation you will be prompted to set the ASNJETPATH environment variable to specify the directory where DB2 DPROP for Microsoft Jet can create the log, trace, and password files. The filenames are:
 - *Apply_qual.LOG*. Created by DB2 DPROP for Microsoft Jet.
 - *Apply_qual.TRC*. Created by DB2 DPROP for Microsoft Jet.
 - *Apply_qual.PWD*. Created by you.

DB2 DPROP for Microsoft Jet also creates the target database in this directory if it does not already exist.

 - Register the Microsoft Jet database source in the ODBC Data Source Administration window, if it is not already registered.

Providing End-User Authentication

If the source or control server requires authentication, create a password file.

Creating a password file:

The password file must meet the following criteria:

- Be named *APPLY_QUAL.PWD*

Where: *APPLY_QUAL* is the Apply qualifier, in upper case. You specify a value for the Apply qualifier when you define a subscription.

For example: *DATADIR.PWD*

Where: *DATADIR* is the Apply qualifier of the subscription defined at the control server.

- Reside in the directory that is specified by ASNJETPATH.
- Contain all server-name/password pairs. This enables you to have a different (or the same) password at each server.
- Have one or more records that use the following format:

```
SERVER=server_name PWD=password USER=userid
```

The file cannot include blank lines or comment lines.

For more information about authentication and security, refer to the *DataJoiner Administration Supplement*.

Chapter 27. Installing the Replication Administration Tool and Connecting All Databases

This chapter describes how to install the replication administration tool and complete the setup of your replication environment. In this chapter you:

- Install the replication administration tool on your Windows 95, 98, or NT workstation.
- Configure access from the workstation on which the replication administration tool is installed to each source, target, and control server.
- In the replication administration tool, set replication-administrative preferences.
- From the workstation on which the replication administration tool is installed, bind DB2 utilities and DB2 CLI to all source, target, and control servers.
- Using the replication administration tool, create replication control tables at the source, target, and control servers.
- For the DB2 source server, bind the Capture program to the DB2 source server. Bind the Apply program to each source, target, and control server. Capture triggers, placed at non-IBM databases, need not be bound.
- From where the Apply program is located, set end-user authentication.

Installing the Replication Administration Tool

The replication administration tool includes ObjectREXX as part of the installation. Check to see if you already have ObjectREXX installed and verify the release level. The replication administration tool will replace your downlevel copy of ObjectREXX, or you can choose not to install ObjectREXX and use your existing copy.

If you are installing the replication administration tool on Windows NT and have not installed the replication administration tool as part of the DataJoiner installation process, log into Windows NT with a valid user name that has administrator authority. If you are installing the replication administration tool on Windows 95 or 98, no administrator log on is required. To install the replication administration tool:

1. Insert the DataJoiner CD-ROM into the appropriate drive. Or download DJRA from the Web.¹⁰

10. <http://www.software.ibm.com/data/dpropr>

2. Use the setup program:
 - a. Open the run window by using one of the following methods:
 - From the Program Manager, select **File** —>**Run**.
 - Select the **Start** button, then select the **Run**.
 - b. In the **Command Line** field, type the following command:
`<cdrom_drive>:\SETUP.exe`

where *<cdrom_drive>* is the letter that designates your CD-ROM drive.

- c. Select **OK** on the Run window.

The first Setup window opens.

3. Respond to the setup program's prompts. Online help is available to help you with the remaining steps.
4. When you have completed setup, the replication administration tool appears in the Windows Program directory. To start the replication administration tool:
 - a. Click the **Start** icon.
 - b. Select the **Programs** menu.
 - c. Select the **DataJoiner for Windows** menu.
 - d. Select the **Replication** menu.
 - e. Select **The Replication Administration Tool**. The replication administration tool primary window opens, as shown in Figure 30 on page 219.

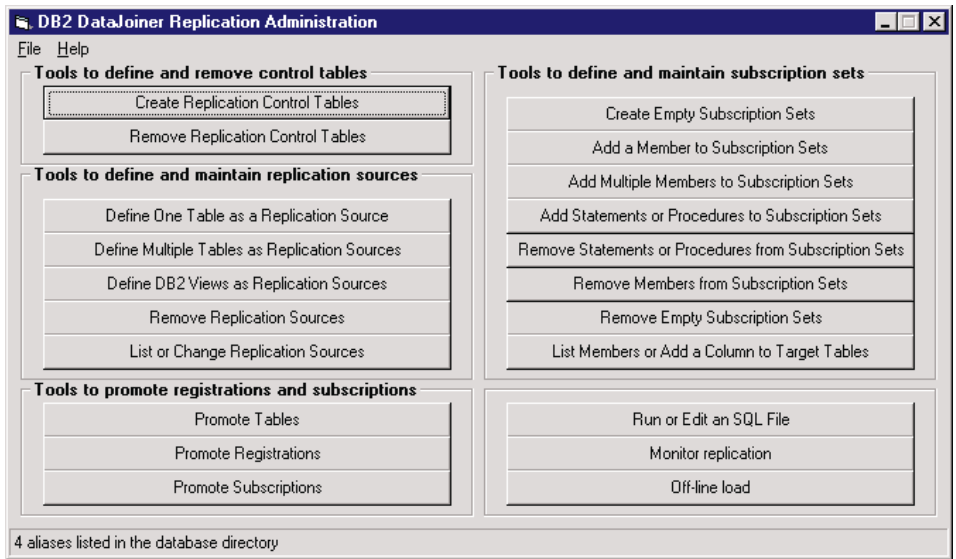


Figure 30. The Replication Administration Tool Primary Window

Configuring Access from the Replication Administration Tool to DataJoiner and DB2

Connect the replication administration tool as a client in DataJoiner to enable the replication administration tool to access source, target, and control servers through DataJoiner. You must connect the replication administration tool to each DataJoiner database that is accessed for replication. Refer to *Installing and Using Clients* for information about configuring clients.

You must also configure database connections from your NT workstation to all DB2 databases involved in replication. Refer to *Installing and Using DB2 Clients for Windows 95 and Windows NT: V2* or the *Installing and Configuring DB2 Clients* for instructions on configuring for DB2 UDB or DB2 client servers. For DB2 for MVS or DB2 for OS/390, refer to “Chapter 19. Configuring Communications for DRDA Data Sources” on page 117, or your *DDCS User’s Guide, V2.3*.

Setting Administrative Preferences

You can specify your preferences for:

- Location of console or file output (optional)
- User IDs and passwords (required)
- Tracing SQL execution activity (optional)

To set preferences, select **Preferences** from the **File** menu on the replication administration tool primary window. You can change the preferences you set whenever you want to.

On the Connection page of the Preferences notebook, you see a list of databases that are currently catalogued on your system. Select a database, then select the **Modify** pushbutton to set the user ID and password.

If you are using Microsoft SQL Server in your replication environment, do not use an alias user ID. Microsoft SQL Server will reject the alias user ID.

Figure 31 shows the Connection page on the Preferences notebook.

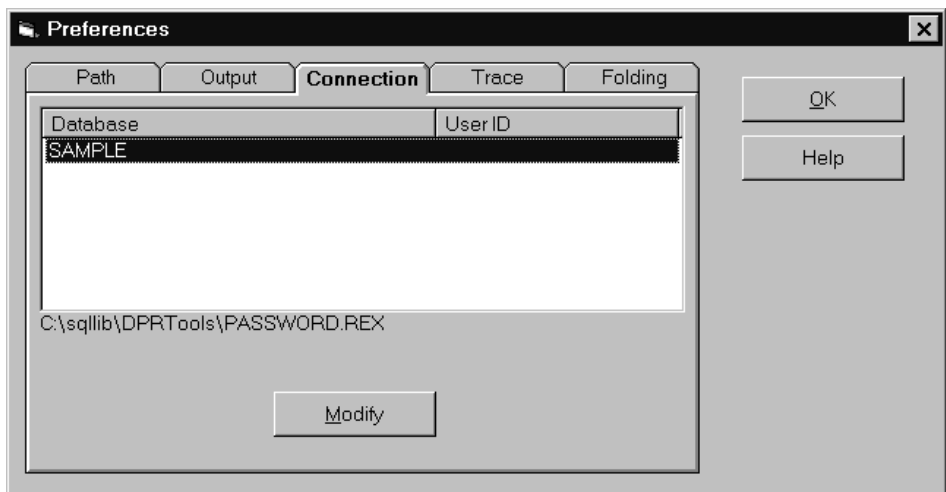


Figure 31. The Preferences Notebook: Connection

Binding to Source, Target, and Control Servers

Bind DB2 utilities to each source, target, and control server, including DataJoiner. You must run the **bind** command separately for each DB2 or DataJoiner source, target, or control server. From the replication administration tool workstation, you must bind DB2 utilities and DB2 CLI; if you have a mixture of Windows NT, Windows 95, and Windows 98 workstations you must bind DB2 utilities to each DataJoiner and each DB2 database from at least one Window 95 or 98 workstation and at least one Window NT workstation. An example of what you enter is:

```
cd c:\sql\lib\bnd
db2 connect to data-server-db
db2 bind @db2ubind.lst blocking all grant public
```

Creating Replication Control Tables

Create control tables at each DataJoiner and DB2 system involved in replication. When you complete this step, the replication administration tool places a register table and a pruning control table at the non-IBM database source and creates nicknames for these tables in DataJoiner.

From the replication administration tool primary window, click the **Create Replication Control Tables**. The created window opens Figure 32.

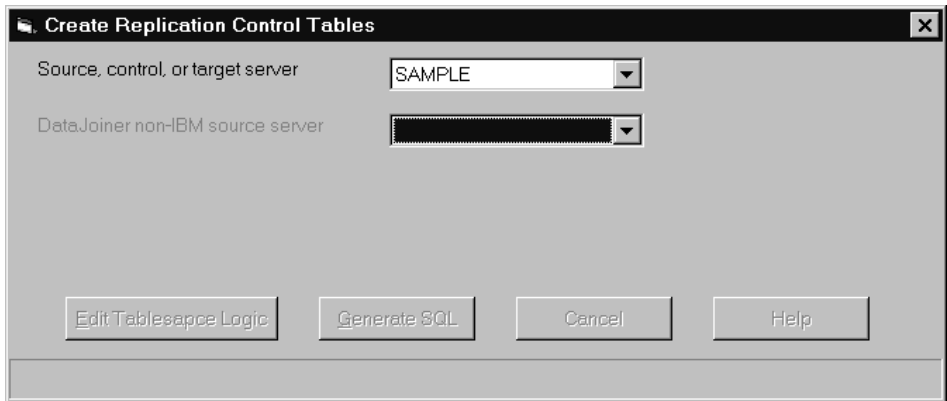


Figure 32. Creating Replication Control Tables.

The fields you complete in order to create a control table are:

Source, control, or target server

When you click the down arrow, the replication administration tool checks to see what type of server it is and then lists all servers available to your user ID and workstation. The list includes DataJoiner servers and DB2 servers. If you select a DataJoiner server from the list, the DataJoiner non-IBM source server pulldown list becomes active. If you do not choose a DataJoiner server, you will link directly to a DB2 target database.

DataJoiner non-IBM source server

If DataJoiner is the replication source server, identify the name of the non-IBM data source that is used in the replication. The pull down list contains DataJoiner server names. The replication administration tool generates the pull down list according to the database name you provided in the **Source, control, or target server** field. Specify **(None)** if you are using the DataJoiner server as a target or control server, but not a source server.

Edit Tablespace Logic

Click on this button to customize table space names for control tables or for CREATE TABLESPACE options. The default table space names for DB2 for OS/390 are:

- TS_UOW for the UOW table
- TS_CNTL for all other control tables

For other platforms, the default table space name is TSnnnnnn, where nnnnnn is a unique identifier.

Generate SQL

Click this push button after you supply all the information on this panel you need to generate SQL. While the SQL is being generated, a window is displayed showing processing messages and error messages, if any.

When the procedure completes successfully, save the file by selecting Save from the File pull down menu. You can now edit the generated SQL, according to the guidelines that are listed in “Editing the Replication Administration Tool-Generated SQL” on page 229. When you are ready, run the SQL by selecting Run from the File pull down menu. You must save the generated SQL before you can run the SQL. You must run the SQL for generating control tables before you generate and run SQL to create replication sources or subscriptions.

Binding the Capture and Apply Programs in DB2 Systems

Before you bind your Capture and Apply programs, make sure that you have created control tables as described in “Creating Replication Control Tables” on page 221.

From the DB2 source server, you must bind the Capture program to the DB2 source server. From each DataJoiner and DB2 source, target, and control server, you must bind the Apply program.

For more information on binding the Apply program in DataJoiner, refer to “Finalizing the Setup for the Apply Program in NT” on page 223. For more information on binding other Capture and Apply programs, refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)*.

Finalizing the Setup for the Apply Program in NT

This section describes how to:

- Bind the Apply program.
- Provide end-user authentication.

Binding the Apply Program in NT

If you create several DataJoiner databases on a DataJoiner instance, you must bind Apply to each of the DataJoiner databases. If you are running the Apply program locally on DataJoiner, you must bind the Apply program to the DB2 database. To bind the Apply program to source, target, and control servers:

1. Ensure that the user ID you are using to bind the Apply program has the required privileges. See “Authorization Requirements” on page 202.
2. Logon to the Windows NT server on which DataJoiner is running using the Apply user ID from the target server system.
3. Change to the directory where the Apply program bind files are located, which is in the DataJoiner instances directory: `home\sqllib\bnd`.
4. Connect to the source server database by entering:

```
DB2 CONNECT TO database
```

where *database* is the source server database.

If the source is a non-IBM database, the Apply program considers DataJoiner as the server database.

5. Create and bind the Apply program packages to the source server database by entering both of the following commands:

```
DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL GRANT PUBLIC
DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL GRANT PUBLIC
```

Where:

CS The list in cursor stability format.

UR

The list in uncommitted read format.

These commands create a list of packages, the names of which can be found in the `APPLYCS.LST` and `APPLYUR.LST` files.

6. Repeat the connect and bind steps for each server that the Apply program connects to. You must bind the Apply program to the source, target, and control servers.

Because the Apply program control tables use static SQL calls, the Apply bind process searches for nearly all of the control tables at each server that it is bound to, regardless of whether or not the control tables are used at each server.

For more information on binding Apply programs, refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)*.

Providing End-User Authentication in NT

Remote DB2 systems involved in replication might require that the Apply program running locally on DataJoiner for NT provide a password or use a user ID different from the ID used on the local DataJoiner server. The Apply password file allows you to connect to the remote DB2 and remote DataJoiner systems. Create the password file on the local DataJoiner system if your remote DB2 and DataJoiner system are cataloged on the local DataJoiner system with AUTHENTICATION=SERVER. If your remote DB2 and DataJoiner system are cataloged with AUTHENTICATION=CLIENT, you do not need to provide a password file. You can also use the password file to use different user IDs on your remote DB2 system than what you use on your local DataJoiner system.

We recommend that you limit read access of this file to the user ID that runs the Apply program.

The password file must:

- Be named:

APPLYQUALinstnameCNTLSRVR.PWD

Where:

APPLYQUAL

The Apply qualifier in uppercase

instname

The DataJoiner instance name that the Apply program runs under. The value of DB2INSTANCE.

CNTLSRVR

The name of the control server, in uppercase

For example: DATADIRapp\y1REPRTDB.PWD.

This naming convention is the same as the .LOG file name and the .SPL file name, but with a file extension of PWD.

- Reside in the directory from which the Apply program starts

- Contain all server-name/user-name/password sets for the file. The user name is optional. The password file enables you to have a different (or the same) password at each server.
- Contain one or more records in the following format:
`SERVER=server_name USER=user_name PWD=password`
 or
`SERVER=server_name PWD=password`
- Do not include blank lines or comment lines.

For more information about authentication and security, refer to the *IBM DataJoiner Administration Supplement* or the *IBM DB2 Universal Database Administration Guide*.

Setting up the NT Service Control Manager

You can operate the Apply program for Windows NT, Windows 95, and Windows 98 by using the DB2 command processor, or by using the NT Service Control Manager (SCM). The SCM enables you to automatically start the Apply program as a service from the NT Control Panel.

To set up NT services:

1. Install the replication service by entering the **ASNINST** command and the fully-qualified path name of the **ASNSERV** executable file, as shown in the following example:
`ASNINST D:\DIRECTORY\ASNSERV.EXE`
2. Set up the service from the NT Control Panel.
 - a. Click the **Services** icon. The Services window opens.
 - b. Select the **Replication** and click the **STARTUP** push button.
 - c. Ensure that the startup type radio button indicates automatic.
 - d. Specify the local user ID and password, then click **OK**. The user ID must be the one that runs the Apply program with the appropriate DB2 privileges.
3. Add the environment variable **ASNPATH** to specify the location of the Apply password file.
 - a. Select **System** from the NT Control Panel. The System Properties window opens.
 - b. Click the **Environment** tab.
 - c. Type the **ASNPATH** string in the **System Variable** field. For example:
`ASNPATH=D:`
 - d. Click **OK**.
4. Create an ASCII file to run the Apply program.

- a. Enter the following records in the file:

db_name x:\ASNAPPLY *parameters*

where:

- *db_name* is the name of your source database and control database for the Apply program.
- *x* is the drive.
- *parameters* is one or more parameters.

For example:

```
DBNAME1 C:\APPLY\ASNAPPLY X 2
```

- b. Save the file to the following name:

D:\NTSERV.ASN

After starting the service, the Apply program runs independently of ASNSERV. Stopping ASNSERV does not stop the Apply program.

To remove ASNSERV from the NT Control Panel, use the ASNREMV program. After removing ASNSERV, the **Replication** no longer appears on the Services window.

Chapter 28. Starting and Using the Replication Administration Tool

This section describes how to begin running the replication administration tool and describes the basic process of setting up replication.

Starting the Replication Administration Tool

To start the replication administration tool:

1. To start the replication administration tool first start all databases involved in replication. Then:
 - a. Click the **Start** icon on the Windows 95, 98, or NT desktop.
 - b. Select the **Programs** menu.
 - c. Select the **DataJoiner for Windows** menu.
 - d. Select the **Replication** menu.
 - e. Select **The Replication Administration Tool**. The replication administration tool primary window opens, as shown in Figure 33.

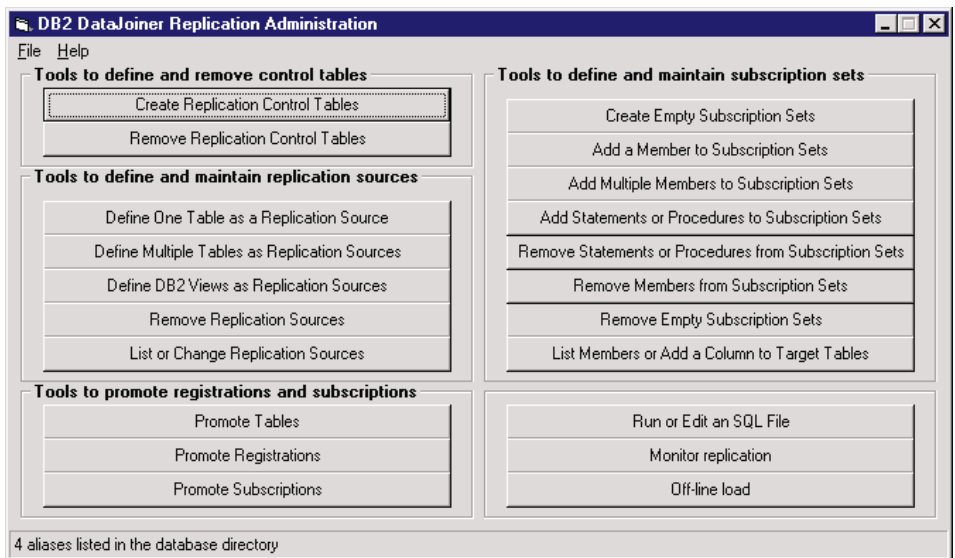


Figure 33. The Replication Administration Tool Primary Window

General Steps for Setting up Replication

Generally, each function in the replication administration tool consists of five steps:

1. Select the function you want (for example, **Define One Table as a Replication Source** or **Add a Member to Subscription Sets**). A window opens.
2. If applicable, edit the replication administration tool logic to affect the statements that are created when you generate SQL by selecting the **Edit Logic**, **Edit Predicate Logic**, or **Edit Create Table Logic** button from the replication administration tool function window. See “Editing the Replication Administration Tool Logic” on page 229 for more information.
3. Generate the SQL.

- a. Provide the required information, as prompted in the displayed window.

Where wildcards are allowed, use either a string with no (%) wildcards, or a pattern string with one or more (%) wildcards, with one trailing (%) wildcard. The (%) symbol matches a substring of none or many arbitrary characters. When a pattern string contains a (%) wildcard, any () characters also in the string are interpreted as single-character wildcards.

- b. Create the SQL by selecting the **Generate SQL** button. SQL is generated into a file but not run yet.
- c. Save the file.

4. Review and edit, if needed, the generated SQL file from the console window or by selecting the **Run or Edit an SQL file** button from the primary window.

DataJoiner’s pass-through and user mapping functions *require* that you use only the replication administration tool to run the SQL. See “Editing the Replication Administration Tool-Generated SQL” on page 229 for more information.

5. Run the SQL file by selecting the **Run or Edit an SQL file** button and then selecting the **Run** button from the primary window.

Make sure that you run your generated SQL files before generating another set of SQL files. If you do not run your SQL after it is generated, the replication administration tool might generate duplicate names for objects. See “Running the Replication Administration Tool-Generated SQL” on page 230 for more information.

Editing the Replication Administration Tool Logic

Three files are available for edit: SRCESVR.REX, TARGSVR.REX, and CNTLSVR.REX.

SRCESVR.REX

Specifies the owner and name of the CD or CCD table and the table space in which the CD or CCD table is placed. Select the function to define one or multiple tables as replication sources and click **Edit Logic** to edit SRCESVR.REX before you generate SQL.

CNTLSVR.REX

Specifies the criteria by which rows from the source table can be replicated to the target. You can specify which columns in a source table are eligible for replication. You also can specify values to search individual source tables and replicate only the data that matches the value criteria. Select the function to add one or multiple members to subscription sets and click **Edit Predicate Logic** to edit the CNTLSVR.REX file.

TARGSVR.REX

Specifies the table space or segment in which to create target tables. Check this file to make sure that the table spaces are being defined in the location of the target database where you want them. Select the function to add one or multiple members to subscription sets and click **Edit Target Table Logic** to edit the TARGSVR.REX file.

TBLSPACE.REX

Specifies the table spaces for the control tables and the UOW table. Check this file to make sure that the table spaces are being defined in the location where you want them. Select the function to create replication control tables and click **Edit Table Logic** to edit the TBLSPACE.REX file.

Note: You can specify where table spaces are created in the SRCESVR.REX and TARGSVR.REX by editing the default directory C:\ to the directory you prefer. When you type in your directory, make sure you add a backslash (\) after the directory. For example, if you are changing the directory from C:\ to F:\Test\, make sure that you enter a slash before the word Test and after the word Test. If you just enter F:\Test, an error will occur when you attempt to generate SQL.

Editing the Replication Administration Tool-Generated SQL

You can edit the replication administration tool's SQL from the console window or by selecting the **Run or Edit an SQL file** button from the primary window. You can edit the SQL for several reasons. For example, you might want to:

- Review and edit the parameters for the CCD tables that are used in the definition of a replication source.

For example, if the two-part names of the CD or CCD tables are altered, the values of columns CD_OWNER, CD_TABLE and CCD_OWNER, CCD_TABLE specified in the ASN.IBMSNAP_REGISTER table row must be changed to match the changes made to the actual table names.

- Edit create table and index statements to represent clusters and other database objects.
- For Oracle, ensure that tables are created in the existing table spaces that you want.
- For Microsoft SQL, create control tables on an existing segment.
- Review and edit subscription member predicates as a way of defining multiple subscriptions at one time. You can use substitute variables in your predicates and resolve the variables with programming logic.

When editing generated SQL, be careful not to change special markers that the replication administration tool places within the SQL. For example, :ENDOFTRIGGER: or :ENDOFPROCEDURE: is part of a comment that is necessary for the replication administration tool to run successfully. Altering create trigger blocks can result in incorrect SQL that ends in error when run.

Running the Replication Administration Tool-Generated SQL

The **Run SQL** button on the replication administration tool GUI is intended to be used for SQL generated by the replication administration tool. SQL you generate outside the replication administration tool might not run successfully if you use the replication administration tool to start it. Likewise, you might not be able to run SQL generated by the replication administration tool at a DB2 command line. We recommend running the replication administration tool-generated SQL through the replication administration tool's GUI.

Running Capture and Apply

After you have defined replication sources and created subscription sets and members, you are ready to run the Capture and Apply programs. For more information on running Capture and Apply programs, refer to the *IBM DB2 Replication Guide and Reference (V5)* or the *IBM DB2 Replication Guide and Reference (V6)*. Capture triggers begin running automatically.

Operating Apply for UNIX platforms

An administrator can use the commands in the following sections to perform the following Apply program tasks:

- Starting

- Scheduling
- Stopping

Before you start the Apply program

Before you start the Apply program, ensure that:

- You have the proper authorization. See *IBM DB2 Replication Guide and Reference (V6)* for information about authorization for the Apply program.
- The control tables are defined.
- At least one subscription is created and activated.
- The Apply package is created.
- A password file has been created, if necessary, for end-user authentication at the source server. See *IBM DB2 Replication Guide and Reference (V6)* for more information.
- The Capture program is started, and the ASN0100I initialization message was issued (if you are running a Capture program).

Starting Apply for UNIX platforms

After you start the Apply program, it runs continuously until:

- You stop it in an orderly way.
- You cancel it.
- An unexpected error or failure occurs.

To start the Apply program on UNIX:

1. Log on with the user ID that will run the Apply program.
2. Ensure that you set the DB2 instance name as shown:


```
export DB2INSTANCE=db2_instance_name
```
3. Set the LIBPATH and language environment variables or edit the .profile file in the same environment in which the Apply program starts.

AIX example:

```
export LIBPATH=db2instance_home_directory/sql1lib/lib:/usr/lib:/lib
export LANG=en_US
```

HP-UX example:

```
export SHLIB_PATH=db2instance_home_directory/sql1lib/lib:/usr/lib:/lib
export LANG=en_US
```

Solaris and UnixWare 7 example:

```
export LD_LIBRARY_PATH=db2instance_home_directory/sql1lib/lib:/usr/lib:/lib
export NLS_PATH=/usr/lib/locale/%L/%N:db2instance_home_directory/sql1lib/msg/en_US/%
export LANG=en_US
```

where *db2instance_home_directory* is the name of the DB2 instance's home directory.

4. Enter the **asnapply** command and options:

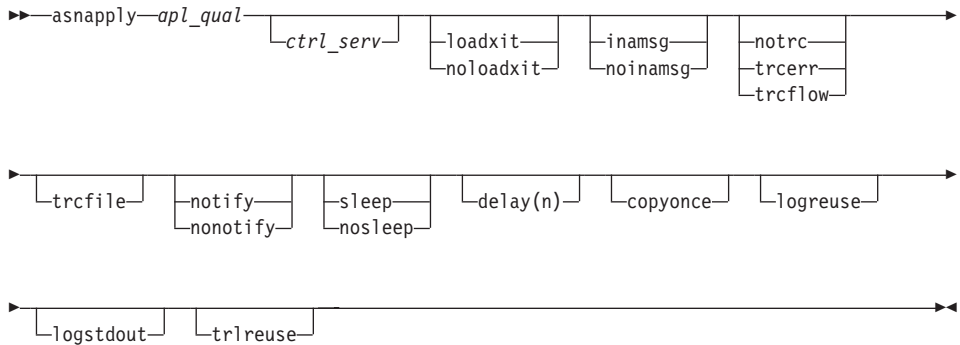


Table 13 defines the invocation parameters.

Table 13. *ASNAPPLY* invocation parameter definitions for UNIX platforms

Parameter	Definition
<i>apl_qual</i>	Specifies the Apply qualifier that the Apply program instance uses to identify the subscriptions to be served. The Apply qualifier is case sensitive and must match the value of APPLY_QUAL in the subscription set table. This must be the first parameter.
<i>ctrl_serv</i>	Specifies the name of the server where the replication control tables will reside. If you do not specify this parameter, the default is the default database or the value of DB2DBDFT.
loadxit	Specifies that the Apply program is to invoke ASNLOAD, an IBM-supplied exit routine that uses the export and load utilities to refresh target tables.
noloadxit (default)	Specifies that the Apply program will not invoke ASNLOAD.
inamsg (default)	Specifies that the Apply program is to issue a message when the Apply program is inactive.
noinamsg	Specifies that the Apply program will not issue this message.
notrc (default)	Specifies that the Apply program does not generate a trace.
trcerr	Specifies that the Apply program generates a trace that contains only error information.

Table 13. ASNAPPLY invocation parameter definitions for UNIX platforms (continued)

Parameter	Definition
trcfLOW	Specifies that the Apply program generates a trace that contains both error and execution flow information.
trcfile	If both trcfLOW and trace are specified, the Apply program writes trace output to the trace file (*.trc). If you do not specify this option, the Apply program sends trace output to the standard output, stdout.
notify	Specifies that the Apply program is to invoke ASNDONE, an exit routine that returns control to the user when the Apply program processing ends.
nonotify (default)	Specifies that the Apply program will not invoke ASNDONE.
sleep (default)	Specifies that the Apply program is to go to sleep if no new subscriptions are eligible for processing.
nosleep	Specifies that the Apply program is to stop if no new subscriptions are eligible for processing.
delay (<i>n</i>)	Where <i>n</i> =0, 1, 2, 3, 4, 5, or 6. Specifies the delay time (in seconds) at the end of each Apply program cycle when continuous replication is used. The default delay time is 6 seconds.
copyonce	The Apply program executes one copy cycle for each eligible subscription set and then terminates. An eligible subscription set is such that: <ul style="list-style-type: none"> • ACTIVATE > 0 • REFRESH_TIMING = R or B or REFRESH_TIMING = E and the specified event has occurred. MAX_SYNCH_MINUTES and END_OF_PERIOD are honored if specified.
logreuse	The Apply program reuses the log file (*.app) by first deleting it and then re-creating it when the Apply program is restarted. If you do not specify this option, the Apply program appends messages to the log file, even after the Apply program is restarted.
logstdout	The Apply program sends all messages to both the standard output (stdout) and the log file.
trlreuse	The Apply program empties the Apply trail table when the Apply program is started.

Scheduling Apply for UNIX platforms

Use the **at** command to start the Apply program at a specific time. For example, the following command starts the Apply program at 3:00 p.m. on Friday:

```
at 3pm Friday asnapply myqual
```

Stopping Apply for UNIX platforms

Use the **asnastop** command or a key combination to stop the Apply program in an orderly way.

▶—asnastop—*apply_qualifier*—▶

To use the command, do the following from a window where the Apply program is not running:

1. Set environment variable **DB2INSTANCE** to the value set when the Apply program was started.
2. Set environment variable **DB2DBDFT** to the source server specified when the Apply program was started (or the **DB2DBDFT** value used when the Apply program was started).
3. Enter the command.

Chapter 29. DataJoiner with the Replication Administration Tool: Data Typing

When you are defining replication sources for non-IBM database tables, data typing often requires extra steps. When the source table is a DataJoiner nickname, DataJoiner handles any data-typing transformations when the nickname of the source table is created outside of the replication administration tool. For more information see the *DataJoiner Application Programming and SQL Reference Supplement*.

During the subscription process, the final data mappings occur. When the target is a table that is accessed through a DataJoiner nickname, the DataJoiner nickname process does not always create the correct type, because:

- DataJoiner requires that the target table be created in the non-IBM database *before* a nickname is created on DataJoiner.
- When the nickname is created, DataJoiner chooses a data type known to DB2 common server that maps to the largest possible value in the target table.
- Sometimes the largest value from the source table (typically in DB2 for MVS), is not the largest value that you can store in the target column.
- The replication administration tool generates ALTER NICKNAME statements for the COLTYPE, SCALE, and LENGTH in the DataJoiner database for the target table nickname. For Oracle, Sybase, and MS SQL Server, the replication administration tool selects the data types in the DataJoiner database. DataJoiner can then perform a conversion when accessing the target table. For Informix, no column modifications are required.

Using the replication administration tool, you can create a subscription in which your non-IBM database is both the source server and target server. When the replication administration tool creates the nickname for the target table, DataJoiner creates the same data types that it generated for the source so that data types match.

In general, you should let the replication administration tool create the target table and ALTER NICKNAME statements. Otherwise, you must take care to choose the same data-type mappings that DataJoiner would generate if you want to create your own target tables on non-IBM databases.

This section discusses how the replication administration tool with DataJoiner handles data typing in three specific scenarios.

DB2 to Oracle Replication

The source is DB2 and the target is Oracle, the following restrictions and conversions are performed:

Table 14. Data Type Conversion: DB2 to Oracle

DB2 for MVS Source data type	Oracle Target data type	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
FOR BIT DATA(n)	RAW(n)			
CHAR (<256)	CHAR(n)			
CHAR (>=256)	VARCHAR2(n)			
VARCHAR2(n)	VARCHAR2(n)			
VARCHAR(n)	VARCHAR2(n)			
GRAPHIC(n)	CHAR(n)			
VARGRAPHIC(n)	VARCHAR(n)			
LONG VARCHAR	LONG			
DATE	DATE	DATE	4	
TIMESTAMP	DATE			
TIME	DATE	TIME	3	
SMALLINT	NUMBER(5)	DECIMAL	5	0
INTEGER	NUMBER(10)	DECIMAL	10	0
DECIMAL(n,m)	DECIMAL(n,m)	DECIMAL	n	m
FLOAT	FLOAT			

DB2 to Informix

When the source is DB2 and the target is Informix, the following restrictions and conversions are performed:

Table 15. Data Type Conversion: DB2 to Informix

DB2 for MVS Source data type	Informix Target data type	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
FOR BIT DATA(n)	CHAR(n) ¹			
CHAR(n)	CHAR(n)			
VARCHAR(<256)	VARCHAR(n)			
VARCHAR(>=256)	TEXT			
GRAPHIC(n)	BYTE			
VARGRAPHIC(n)	BYTE			
				No updates occur after create nickname

Table 15. Data Type Conversion: DB2 to Informix (continued)

DB2 for MVS Source data type	Informix Target data type	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
LONG	TEXT			
VARCHAR				
DATE	DATE			
TIMESTAMP	DATETIME			
	YEAR TO			
	FRACTION(5)			
TIME	DATETIME			
	HOUR TO			
	SECOND			
SMALLINT	SMALLINT			
INTEGER	INT			
DECIMAL(n,m)	DECIMAL(n,m)			
FLOAT	FLOAT			

Note: ¹For CHAR data in Informix, a string is terminated when the first non-printable character is encountered. In this example, the CHAR FOR BIT data from DB2/MVS could be truncated when stored in Informix, making it appear as though an x'00' is part of the string.

DB2 to Microsoft SQL Server, Sybase, or Sybase SQL Anywhere

When the source is DB2 and the target is MS SQL Server, Sybase, or Sybase SQL Anywhere, the following restrictions and conversions are performed:

Table 16. Data Type Conversion: DB2 to MS SQL Server, Sybase, or SQL Anywhere

DB2 for MVS SOURCE	MS SQL SERVER, SYBASE, OR SQL ANYWHERE TARGET	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
CHAR (n) FOR BIT DATA	BINARY(n)	CHAR ¹		
VARCHAR(n) FOR BIT DATA(N)	VARBINARY(n) or BINARY(n)			
CHAR(n)	CHAR(n)			
VARCHAR(<256)	VARCHAR2(n) or VARCHAR(n)			
VARCHAR(<256)	TEXT			
GRAPHIC(n) ²	?			

Table 16. Data Type Conversion: DB2 to MS SQL Server, Sybase, or SQL Anywhere (continued)

DB2 for MVS SOURCE	MS SQL SERVER, SYBASE, OR SQL ANYWHERE TARGET	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
VARGRAPHIC(n)? 2				
LONG VARCHAR 2	?			
DATE	DATETIME or DATE	DATE	4	
TIMESTAMP	DATETIME or DATESTAMP			
TIME	DATETIME or TIME	TIME	3	
SMALLINT	SMALLINT			
INTEGER	INT			
DECIMAL(n,m)	DECIMAL(n,m)			
FLOAT	FLOAT			

Note: ¹The replication administration tool creates the target table in MS SQL Server with data type "binary." The DataJoiner nickname is created with COLTYPE of VARCHAR. The replication administration tool updates the COLTYPE to CHAR.

Note: ²These DB2/MVS data types have not been tested.

DB2 to Microsoft Jet

When the source is DB2 and the target is Microsoft Jet, the following restrictions and conversions are performed:

Table 17. Data Type Conversion: DB2 to Microsoft Jet

DB2 for MVS SOURCE	MS JET TARGET	Changes after Create Nickname		
		COLTYPE	LENGTH	SCALE
CHAR (n) FOR BIT DATA	Memo or OLE Object ¹			
VARCHAR(n) FOR BIT DATA(N)	Memo or OLE Object ¹			
CHAR(n)	Text or Memo ¹			
VARCHAR(<256)	Text or Memo ¹			
GRAPHIC(n) 2	?			
VARGRAPHIC(n)? 2				

Table 17. Data Type Conversion: DB2 to Microsoft Jet (continued)

DB2 for MVS	MS JET	Changes after Create Nickname		
SOURCE	TARGET	COLTYPE	LENGTH	SCALE
LONG	Memo			
VARCHAR ²				
DATE	Datetime			
TIMESTAMP	Datetime			
TIME	Text			
SMALLINT	Number (Integer)			
INTEGER	Number (Long Integer)			
DECIMAL(n,m), NUMERIC	Number (Integer), Number (Long Integer), Number (Double), or Text ¹			
PRECISION, FLOAT, DOUBLE	Number (Double)			

Note: ¹The value used depends on the precision set for the field.

Note: ²These DB2/MVS data types have not been tested.

Chapter 30. Non-IBM Databases: Table Names, Control Tables, and the CCD

This section describes how the replication control, source, and CCD table names in a non-IBM database and their associated nicknames are determined.

Table Names and Qualifiers

Both a table and a DataJoiner nickname must exist for any non-IBM table that you attempt to define as a replication source. The unqualified table and DataJoiner nicknames must match, though the table and nickname qualifiers can differ.

Control Tables in a Non-IBM Database

The Apply program refers to the register control table and pruning control table by using ASN as the qualifier. Creating tables with the ASN qualifier in the non-IBM database assumes that an ASN schema exists and that the person creating the control tables has authority to create tables in the ASN schema. To compensate for the assumptions, when you select **Create Replication Control Tables**, the replication administration tool creates the control tables in the non-IBM database using the remote ID stored in DataJoiner of the user as the table qualifier, but always using ASN as the DataJoiner nickname qualifier for the control tables.

CCD Table and Capture Triggers

There are two parts to the CCD table name: the unqualified name and the qualifier.

The default name is CCD followed by a series of character-digits. The default digit-string is derived from a timestamp. You can set the CCD table name by assigning a string to OUT.CD_TABLE (this program variable is used for either CD or CCD table names) in the SRCESVR.REX file.

The default CCD qualifier is the table qualifier (not nickname qualifier) of the non-IBM source table. You can set the qualifier by assigning a string to OUT.CD_OWNER in the SRCESVR.REX file. Use caution when changing the CCD qualifier because triggers perform inserts to the CCD tables, and they are created using the register and pruning control table qualifiers; you have to

ensure the triggers have the proper authorizations. IBM recommends not changing the CCD qualifier for non-IBM databases.

The Capture and pruning triggers refer to and use the qualifiers for the register control table and pruning control table, not the ASN nickname qualifiers. Capture triggers, referring to source tables and the CCD tables, use the table qualifiers, not the DataJoiner nickname qualifiers.

CCD Nicknames in DataJoiner

The default CCD nickname qualifier is the qualifier of the nickname for the non-IBM source table. If `OUT.CD_OWNER` is set in the `SRCESVR.REX` file, the setting overrides the default and is used as the CCD nickname qualifier. The unqualified name of the CCD table (which you can set in `SRCESVR.REX`) uses the unqualified portion of the DataJoiner nickname for the CCD table.

The Register and Pruning Control Tables

The `ASN.IBMSNAP_REGISTER` and `ASN.IBMSNAP_PRUNCNTL` nicknames refer to tables in the non-IBM database with the same unqualified names with probably a different qualifier, for example, `MYUSERID.IBMSNAP_REGISTER`. Even though the register and pruning control tables reside in a non-IBM database, references to the source and CCD tables in the register control table rows use the nicknames, not the table names, so that the Apply process operates entirely in the DataJoiner name space. The Apply process is not aware that a particular source table is a table in a non-IBM database.

Part 6. Appendixes

Appendix A. DRDA Application Server

Using DataJoiner as a DRDA AS enables DataJoiner to function as a database server for Application Requesters that use the DRDA protocol, as well as for other clients that use DB2 private protocols; the AS option provides support for DRDA level 1. With the AS option installed, applications that use DRDA Application Requesters (AR) can create packages at the DataJoiner server and subsequently execute them subject to the SQL that is supported at the server.

For more information about DRDA concepts and a detailed description of DRDA commands and bind options, refer to the following publications:

- *Distributed Relational Database Architecture Reference* (SC26-4651-01)
- *DB2 for MVS Connections with AIX or OS/2* (SG24-4558-00)

Supported DRDA Functions

In DRDA, a function is categorized as either required or optional. Table 18 identifies which functions (required and optional) are implemented in the AS on DataJoiner, and Table 19 on page 246 lists the bind option values that are supported.

Table 18. DataJoiner DRDA Application Server Function

Description	Required (R) Optional (O)	Supported	Not Supported
DRDA level 1 required function	R	X (Note 1)	
Rebind	O		X
Describe user privileges	O		X
Describe RDB table	O		X
Interrupt RDB request	O		X

Note:

(1) Certain required functions are not supported.

Bind Options Supported by the DataJoiner DRDA Application Server

Table 19 on page 246 shows the BIND options that are supported by the DataJoiner DRDA Application Server.

Table 19. Bind Options Supported by the DataJoiner DRDA Application Server

Bind Option	Value	Supported?	DB2 for MVS Precompile Option (Note 1)	DB2/VM Pre-processing Option	OS/400 Precompile Option	DDCS Prep or Bind Option
Package Version Name	<u>Null</u>	No	VERSION			VERSION
	Any other value	No				
Bind Existence Checking	<u>Object existence optional</u>	No	VALIDATE (<u>RUN</u>) ^b	<u>NOEXIST</u>	GENLVL(<u>10</u> , 11-40)	VALIDATE <u>RUN</u>
	Object existence required	Yes	VALIDATE (BIND) ^b	EXIST	GENLVL(00-09)	VALIDATE BIND
Package Replacement Option	<u>Replacement allowed</u>	No	ACTION (<u>REPLACE</u>)	<u>REPLACE</u>	REPLACE (<u>*YES</u>)	ACTION <u>REPLACE</u>
	Replacement not allowed	No	ACTION (ADD)	NEW	REPLACE (*NO)	ACTION ADD
Package Authorization Option	<u>Keep authorizations</u>	Yes		<u>KEEP</u>		RETAIN <u>YES</u>
	Revoke authorizations	No		REVOKE		RETAIN NO
Statement String Delimiter (Note 2)	Apostrophe	Yes	<u>APOSTSQL</u>	<u>SQLAPOST</u>	OPTION([...] *APOSTSQL) (Note 3)	STRDEL APOSTROPHE
	Double quote	No	QUOTESQL	SQLQUOTE	OPTION([...] *QUOTESQL) (Note 4)	STRDEL QUOTE
Statement Decimal Delimiter (Note 5)	Period	Yes	<u>PERIOD</u>	<u>PERIOD</u>	OPTION([...] *PERIOD) or OPTION([...] *SYSVAL) (Note 6)	DECDEL PERIOD
	Comma	No	COMMA	COMMA	OPTION([...] *COMMA) or OPTION([...] *SYSVAL) (Note 6)	DECDEL COMMA

Table 19. Bind Options Supported by the DataJoiner DRDA Application Server (continued)

Bind Option	Value	Supported?	DB2 for MVS Precompile Option (Note 1)	DB2/VM Pre-processing Option	OS/400 Precompile Option	DDCS Prep or Bind Option
Date Format (Note 7)	<u>ISO</u>	Yes	DATE (ISO) (Note 8)	DATE (<u>ISO</u>)	DATFMT (*ISO) (Note 8)	DATETIME <u>ISO</u> (Note 9)
	USA	Yes	DATE (USA)	DATE(USA)	DATFMT (*USA)	DATETIME USA
	EUR	Yes	DATE (EUR)	DATE (EUR)	DATFMT (*EUR)	DATETIME EUR
	JIS	Yes	DATE (JIS)	DATE (JIS)	DATFMT (*JIS)	DATETIME JIS
Time Format (Note 7)	<u>ISO</u>	Yes	TIME (ISO) (Note 8)	TIME (<u>ISO</u>)	TIMFMT (*ISO) (Note 8)	DATETIME <u>ISO</u> (Note 9)
	USA	Yes	TIME (USA)	TIME (USA)	TIMFMT (*USA)	DATETIME USA
	EUR	Yes	TIME (EUR)	TIME (EUR)	TIMFMT (*EUR)	DATETIME EUR
	JIS	Yes	TIME (JIS)	TIME (JIS)	TIMFMT (*JIS)	DATETIME JIS
Package Isolation Level (Note 10)	Repeatable read	Yes	ISOLATION (<u>RR</u>) ^b	ISOLATION (<u>RR</u>)		ISOLATION RR
	Read Stability (All)	Yes		ISOLATION (RS)	COMMIT (*ALL)	ISOLATIONRS
	Cursor stability	Yes	ISOLATION (CS) ^b	ISOLATION (CS)	COMMIT (*CS)	ISOLATION <u>CS</u>
	Uncommitted Read (Change)	Yes		ISOLATION (UR)	COMMIT (<u>*CHG</u>)	ISOLATION UR
	No commit	No (Note 11)			COMMIT (*NONE)	ISOLATION NC
Bind Creation Control	<u>No errors allowed</u>	Yes	SQLERROR (<u>NOPACKAGE</u>) ^b	<u>NOCHECK</u>	OPTION([...] *GEN) GENLVL(00-09, <u>10</u> , 11-20)	SQLERROR <u>NOPACKAGE</u>
	Check only	Yes		CHECK	OPTION([...] *NOGEN)	SQLERROR CHECK

Table 19. Bind Options Supported by the DataJoiner DRDA Application Server (continued)

Bind Option	Value	Supported?	DB2 for MVS Precompile Option (Note 1)	DB2/VM Pre-processing Option	OS/400 Precompile Option	DDCS Prep or Bind Option
	Errors allowed	No	SQLERROR (CONTINUE) ^b	ERROR	OPTION([...] *GEN) GENLVL(21-40)	SQLERROR CONTINUE
Bind Explain Option	No SQL statements	Yes	EXPLAIN (NO) ^b	EXPLAIN (NO)		EXPLAIN <u>NO</u>
	All explainable SQL statements	No	EXPLAIN (YES) ^b	EXPLAIN (YES)		EXPLAIN YES
Package Owner Identifier	< Authorization ID >	No	OWNER ^b	OWNER		OWNER
	Any other value	No				
RDB Release Option	Release at commit	No	RELEASE (COMMIT) ^b	RELEASE (COMMIT)		RELEASE <u>COMMIT</u>
	Release at conversation deallocation	No	RELEASE (DEALLOCATE) ^b	RELEASE (DEALLOCATE)		RELEASE DEALLOCATE
Default RDB Collection ID	< Authorization ID >	No	QUALIFIER ^b	QUALIFIER	DFTRDBCOL	QUALIFIER
	Any other value	No				
Title (Package Description)	Any value (ignored by DB2)	No		LABEL	TEXT	TEXT
Query Block Protocol Control	Fixed row	Yes	CURRENTDATE ^b	NOBLOCK	ALWBLK (* <u>READ</u>)	BLOCKING <u>UNAMBIG</u>
	Limited block	Yes	CURRENTDATE ^b	NOBLOCK	ALWBLK (*ALLREAD)	BLOCKING ALL
	Forced fixed row	Yes		<u>NOBLOCK</u>	ALWBLK (*NONE)	BLOCKING NO
Package Default Char. Subtype						
	Use system default	No				CHARSUB <u>DEFAULT</u>
If Default CCSID is SBCS	BIT	No		CHARSUB (BIT)		CHARSUB BIT

Table 19. Bind Options Supported by the DataJoiner DRDA Application Server (continued)

Bind Option	Value	Supported?	DB2 for MVS Precompile Option (Note 1)	DB2/VM Pre-processing Option	OS/400 Precompile Option	DDCS Prep or Bind Option
If Default CCSID is SBCS	SBCS	No		CHARSUB (SBCS)		CHARSUB SBCS
If Default CCSID is SBCS	MBCS	No		CHARSUB (MBCS)		CHARSUB MBCS
If Default CCSID is MBCS	BIT	No		CHARSUB (BIT)		CHARSUB BIT
If Default CCSID is MBCS	SBCS	No		CHARSUB (SBCS)		CHARSUB SBCS
If Default CCSID is MBCS	MBCS	No		CHARSUB (MBCS)		CHARSUB MBCS
	Any other value	No				
Package Default CCSID	Value specified when DB2 database was created	No		CCSIDSBCS() CCSIDGRAPHIC() CCSIDMIXED()		CCSID CCSIDG CCSIDM
	Any other value	No				
Decimal Precision (Note 12)	31	No	DEC (31)			DEC 31
	Any other value	No	DEC (15)			DEC 15
Replaced Package Version Name	Null	Yes	REPLVER ^b			REPLVER
	Any other value	No				

Table 19. Bind Options Supported by the DataJoiner DRDA Application Server (continued)

Bind Option	Value	Supported?	DB2 for MVS Precompile Option (Note 1)	DB2/VM Pre-processing Option	OS/400 Precompile Option	DDCS Prep or Bind Option
-------------	-------	------------	----------------------------------------	------------------------------	--------------------------	--------------------------

Notes:

- (*) Default values are in **bold**.
- (1) Most are precompile options. Bind options are indicated by ^b.
- (2) Defaults to what the target database supports. For DataJoiner the default is apostrophe.
- (3) Default for non-COBOL applications.
- (4) Default for COBOL applications.
- (5) Defaults to what the target database supports. For DataJoiner the default is period.
- (6) Depending on the installation, *SYSVAL is equivalent to *PERIOD or *COMMA.
- (7) Date and time formats must be the same for the DataJoiner DRDA AS.
- (8) Default is dependent on the installation.
- (9) Format applies to both date and time. If not specified, it defaults based on the country code. This default is mapped to ISO in DRDA flow.
- (10) Package isolation level has no default because an explicit value is always present in the DRDA datastream.
- (11) The isolation level will be escalated to Uncommitted Read (Change).
- (12) Defaults to what the target database supports. For DataJoiner the default is 31.

Restrictions

The database that is being connected to cannot be cataloged on the DataJoiner DRDA AS node as remote (you can obtain this information by using the LIST DATABASE DIRECTORY command). Any attempt to access a remote database results in SQLCODE -1334 being returned on the connect.

Refer to Table 18 on page 245 and Table 19 on page 246 for a list of DRDA options that are not supported by the DataJoiner DRDA Application Server.

The following required DRDA functions are not supported by this DRDA Application Server:

- Object Existence Optional (BNDEXSOPT)
This is the default for the Bind Existence Checking (BNDCHKEXS) bind option. DataJoiner does not support this particular bind option value.
- Multiple binds in a logical unit of work (LUW)
This corresponds to DRDA rules PB5 and PB6.

All input character data is converted by the DRDA Application Server to the database default CCSID (except when the data is tagged as BIT) before it gets to DataJoiner. As a result, you should be aware of the following significant restriction when using this DRDA Application Server:

- Always use input character data that is tagged as BIT if you do not want it converted to the database default CCSID for the operation.

This restriction applies even if you are performing an operation (for example, UPDATE) against a character column that is defined as FOR BIT DATA. If you do not tag the input data as BIT, the results are unpredictable.

Configuration Considerations

You configure the DRDA Application Server using the *drda_heap_sz* (DRDA Heap Size) database manager configuration parameter, which specifies the amount of memory, in pages, that is allocated for use by the DRDA Application Server.

CCSID Support

The following list explains each column of Table 20, which follows.

- **Database Code Page** shows the IBM-defined code page as mapped from the operating system code set.
- **Country Name** specifies the name of the country. Where required, the language for that locale is listed in parentheses ().
- **EBCDIC AR CCSID** shows the CCSIDs that are supported for the EBCDIC AR applications.
- **ASCII AR CCSID** shows the CCSIDs that are supported for the ASCII AR applications.

Table 20. CCSID Support for DataJoiner for Windows NT DRDA Application Server

Database Code Page	Country Name	EBCDIC AR CCSID	ASCII AR CCSID
437 1252	Australia Belgium Canada (English) Finland France Germany Italy Latin America Netherlands Spain Sweden Switzerland UK USA	037 273 277 278 280 284 285 297 500 871	437 819 850
850 1252	Arabic Countries Australia Belgium Canada (English) Canada (French) Denmark Finland France Germany Italy Latin America Netherlands Norway Portugal Spain Sweden Switzerland UK USA	037 273 277 278 280 284 285 297 500 871	437 819 850
850 1250	Czech Republic Croatia Hungary Poland Romanic Serbia/Montenegro (Latin) Slovakia Slovenia	870	852 912

Table 20. CCSID Support for DataJoiner for Windows NT DRDA Application Server (continued)

Database Code Page	Country Name	EBCDIC AR CCSID	ASCII AR CCSID
855 1251	Bulgaria FYR Macedonia Serbia/Montenegro (Cyrillic)	1025	852 912
855 1255	Turkey	1026	857 920
862 1255	Israel	424	862 916
863 1252	Canada (French)	037 273 277 278 280 284 285 297 500 871	863
864 1256	Arabic Countries	420	864 1046 1089
866 1251	Russia	1025	866 915
869 1253	Greece	875	813 869
874	Thailand	838	874
932 948 950	Japan	930 939 5026 5035	932 942
938 948 950	Taiwan	937	938 948 950
949	Korea	933	934 944 949 970
1381	PRC (China)	935	1381 1383

Supported DRDA Application Requesters

The following sections describe the IBM ARs that are supported. For information about other ARs, contact your supplier.

MVS DRDA Application Requesters

The PTFs required for an MVS AR depend on the version of DB2 for MVS that you are using:

- DB2 for MVS Version 3.1 AR requires PTFs UN75959, UN80493, and UN80607 to be applied
- DB2 for MVS Version 4.1 AR requires PTFs UN80494 and UN80608 to be applied

DDCS DRDA Application Requesters

You can use DDCS Version 2.3 ARs with the DataJoiner DRDA AS. You do not have to apply any PTFs.

DB2 for VM DRDA Application Requesters

The PTFs required for a DB2 for VM (SQL/DS) AR depend on the version of DB2 for VM that you are using:

- DB2 for VM Version 3.3 AR requires PTF UN47865 to be applied
- DB2 for VM Version 3.4 AR does not require any PTFs to be applied

Additional steps are required to ensure that the following DB2 for VM utilities work properly against a DataJoiner database through the DRDA Application Server:

- SQLDBSU
 1. Ensure that the correct PTF is installed on your DB2 for VM system:
 - DB2 for VM V3R3 requires PTF UN84183 to be installed.
 - DB2 for VM V3R4 requires PTF UN84184 to be installed.
 2. Create dummy tables in your DataJoiner database by executing the sqldbsu utility that is provided with DataJoiner as follows:


```
sqldbsu database_name
```
 3. Bind SQLDBSU from DB2 for VM. See the description of using a DRDA environment in the *SQL/DS System Administration for IBM VM Systems* book for details. (You can skip the step on creating and populating the SQLDBA.DBSOPTIONS table because this is done by the sqldbsu utility in the previous step.)
- ISQL
 1. Complete the steps that are described above for SQLDBSU.
 2. Set up dummy tables in your DataJoiner database by executing the isql utility that is provided with DataJoiner as follows:


```
isql database_name
```
 3. Bind ISQL from DB2 for VM. See the description of using a DRDA environment in the *SQL/DS System Administration for IBM VM Systems* book for details.

Note: The sqldbsu and isql utilities are in \$HOME/sql1lib/misc where \$HOME is the home directory of the instance owner.

No special setup is required for RXSQL; see the *SQL/DS Procedures Language Interface Installation* book for details.

DB2 for AS/400 DRDA Application Requesters

For OS/400 DRDA ARs to work against the DataJoiner DRDA AS, PTFs need to be applied on OS/400 as follows:

Table 21. OS/400 PTFs required for DRDA AS

OS/400 PTR#	OS/400 V2R3 PTF	OS/400 V3R0.5 PTF	OS/400 V3R1 PTF
SA42461	SF23100	SF23950	SF23270
SA43024	SF23205	SF23950	SF23277

Table 21. OS/400 PTFs required for DRDA AS (continued)

OS/400 PTR#	OS/400 V2R3 PTF	OS/400 V3R0.5 PTF	OS/400 V3R1 PTF
3989448	SF23101	SF23994	SF23271
3993315	SF23722	SF23988	SF23721
3994088	SF23987	SF23986	SF23985
3994297	SF23990	SF23989	SF23960

Appendix B. Configuration Parameters

DataJoiner has been designed with an extensive array of tuning parameters and configuration parameters. These parameters fall into two categories:

- Database manager parameters
- Database parameters

Descriptions for all the configuration parameters are available in the *DB2 Administration Guide*.

The default values and the range values for several of the configuration parameters are dependent on the operating system on which the database manager is installed. In addition, some database manager configuration parameters are applicable only in specific operating environments.

The following table lists the database manager configuration parameters and the default values and valid ranges for DataJoiner. Configuration parameters that are not used on this platform are indicated by the string: "Not applicable for DataJoiner for Windows NT systems."

Note: All the units in this table are in KBs.

Table 22. Database manager configuration parameters, default values, and valid ranges for DataJoiner

Parameter	Description	Default [Range]
<i>adsm_nodename</i>	ADSM node name	Null [any valid string]
<i>adsm_owner</i>	ADSM owner	Null [any valid string]
<i>adsm_password</i>	ADSM password	Null [any valid string]
<i>agent_stack_sz</i> a	Agent stack size	16 [8 - 1000]
<i>agentpri</i>	Priority of Agents	3 (normal) [1 - 6]
<i>aslheapsz</i>	Application support layer heap size	15 [1 - 524288]
<i>authentication</i>	Authentication type	SERVER [CLIENT, SERVER, DCS]
<i>backbufsz</i>	Default Backup Buffer Size	1024 [16 - 524288]
<i>cpuspeed</i>	CPU Speed	-1 [-1, 1e-10 - 1]
<i>dft_account_str</i>	Default accounting string	Null [any valid string]
<i>dft_client_comm</i>	Default client communications protocol	Not applicable for DataJoiner for Windows NT systems

Table 22. Database manager configuration parameters, default values, and valid ranges for DataJoiner (continued)

Parameter	Description	Default [Range]
<i>dft_mon_bufpool</i>	Default value of snapshot monitor buffer pool switch	OFF [ON, OFF]
<i>dft_mon_lock</i>	Default value of snapshot monitor lock switch	OFF [ON, OFF]
<i>dft_mon_sort</i>	Default value of snapshot monitor sort switch	OFF [ON, OFF]
<i>dft_mon_stmt</i>	Default value of snapshot monitor statement switch	OFF [ON, OFF]
<i>dft_mon_table</i>	Default value of snapshot monitor table switch	OFF [ON, OFF]
<i>dft_mon_uow</i>	Default value of snapshot monitor unit-of-work switch	OFF [ON, OFF]
<i>dftdbpath</i>	Default database path	drive on which DataJoiner for Windows NT is installed [any valid drive letter]
<i>diaglevel</i>	Diagnostic error capture level	3 [0 - 4]
<i>diagpath</i>	Diagnostic data directory path	Null [any valid path]
<i>dir_cache</i>	Directory cache support	OFF [ON,OFF]
<i>dir_obj_name</i>	Object name in DCE namespace	Not applicable for DataJoiner for Windows NT systems
<i>dir_path_name</i>	Directory path name in DCE namespace	Not applicable for DataJoiner for Windows NT systems
<i>dir_type</i>	Directory services type	NONE [NONE]
<i>dos_rqrioblk</i>	DOS requester I/O block size	Not applicable for DataJoiner for Windows NT systems
<i>drda_heap_sz</i>	DRDA heap size	128 [16 - 60000]
<i>fileserv</i>	IPX/SPX fileserv name	Null [any valid string]
<i>indexrec</i>	Index re-creation time	access [restart, access]
<i>ipx_socket</i>	IPX/SPX socket number	879E [879E - 87A2]
<i>keepdari</i>	Keep DARI process indicator	Yes [Yes, No]
<i>max_idleagents</i>	Maximum number of idle agents	3 [0 - maxagents]
<i>maxagents</i>	Maximum number of agents	200 [1 - 64000]
<i>maxcagents</i>	Maximum number of concurrent agents	-1 (maxagents) [-1, 1 - maxagents]

Table 22. Database manager configuration parameters, default values, and valid ranges for DataJoiner (continued)

Parameter	Description	Default [Range]
<i>maxdari</i>	Maximum number of DARI processes	-1 (maxagents) [-1, 0 - maxagents]
<i>maxtotfilop</i>	Maximum total files open per application	16000 [100 - 32768]
<i>min_priv_mem</i>	Minimum committed private memory	32 [32 - 112000]
<i>mon_heap_sz</i>	Database system monitor heap size	Database Server with local clients: 12 [0 - 60000] Database Server with local and remote clients: 24 [0 - 60000]
<i>nname</i>	NetBIOS node name	Null [any valid string]
<i>numbd</i>	Maximum number of concurrently active databases	Database Server with local clients: 3 [1 - 60000] Database Server with local and remote clients: 8 [1 - 60000]
<i>objectname</i>	IPX/SPX database manager object name	Null [any valid string]
<i>priv_mem_thresh</i>	Private memory threshold	1296 [-1, 32 - 112000]
<i>query_heap_sz</i>	Query heap size	1000 [2 - 524288]
<i>restbufsz</i>	Default restore buffer size	1024 [16 - 524288]
<i>resync_interval</i>	Transaction resynchronization interval	180 [60 - 60000]
<i>route_obj_name</i>	Routing information object name	Not applicable for DataJoiner for Windows NT systems
<i>rqioblk</i>	Client I/O block size	32767 [4096 - 65535]
<i>sheapthres</i>	Sort heap threshold	2048 [250 - 524288]
<i>sqlstmtsz</i>	SQL statement size	256 [0 - 32767]
<i>svcname</i>	TCP/IP service name	Null [any valid string]
<i>sysadm_group</i>	System administration authority group name	Null [any valid group name]
<i>sysctrl_group</i>	System control authority group name	Null [any valid group name]
<i>sysmaint_group</i>	System maintenance authority group name	Null [any valid group name]
<i>tm_database</i>	Transaction manager database name	Null [any valid string]

Table 22. Database manager configuration parameters, default values, and valid ranges for DataJoiner (continued)

Parameter	Description	Default [Range]
<i>tp_mon_name</i>	Transaction processor monitor name (for all UNIX environments)	Not applicable for DataJoiner for Windows NT systems
<i>tpname</i>	APPC transaction program name	Null [any valid string]
<i>udf_mem_sz</i>	UDF shared memory set size	256 [128 - 60000]

- a** AGENT_STACK_SIZE represents the amount of the agent's stack that is committed to memory when the agent is created. The maximum size by which an agent's stack can grow is determined as follows:
- If the agent stack size is less than or equal to 256 KB, the agent stack can grow to a maximum of 256 KB.
 - If the agent stack size is greater than 256 KB, the maximum agent stack size is rounded up to the nearest 1MB boundary.

If the stack requirements exceed the maximum agent stack size, the DB2 System Controller will terminate abnormally.

The following table lists the database configuration parameters whose defaults or ranges are influenced by the operating system on which the database manager was installed. The default values and valid ranges are given. Configuration parameters that are not used on this platform are indicated by the string: "Not applicable for DataJoiner for Windows NT systems". This table does not include parameters that cannot be modified. (See the *DB2 Administration Guide* for a complete list of parameters, including those provided for information purposes.

Table 23. Database manager configuration parameters whose defaults or ranges are influenced by the operating system on which the database manager was installed

Parameter	Description	Default [Range]
<i>applheapsz</i>	Application Heap Size	128 [32 - 60000]
<i>autorestart</i>	Auto Restart Enable	ON [ON, OFF]
<i>avg_appls</i>	Average number of active applications	1 [1 - <i>maxappls</i>]
<i>buffpage</i>	Buffer pool size	250 [2* <i>maxappls</i> - 524288]
<i>catalogcache_sz</i>	Catalog Cache Size	Database Server with local clients: 16 [1 - <i>dbheap</i>] Database Server with local and remote clients: 32 [1 - <i>dbheap</i>]
<i>chnpggs_thresh</i>	Changes pages threshold	60 [5 - 80]

Table 23. Database manager configuration parameters whose defaults or ranges are influenced by the operating system on which the database manager was installed (continued)

Parameter	Description	Default [Range]
<i>copyprotect</i>	Copy Protection Enable	No [No, Yes]
<i>dbheap</i>	Database Heap	Database Server with local clients: 300 [32 - 60000] Database Server with local and remote clients: 600 [32 - 60000]
<i>dft_extent_sz</i>	Default extent size of table spaces	32 [2 - 256]
<i>dft_loadrec_ses</i>	Default number of load recovery sessions	1 [1 - 30000]
<i>dft_prefetch_sz</i>	Default prefetch size	16 [0 - 32767]
<i>dir_obj_name</i>	Object name in DCE Namespace	Null [any valid string]
<i>dlchktime</i>	Time interval for checking deadlocks	10000 [1000 - 600000]
<i>indexrec</i>	Index recreation time	Use system setting [system; access; restart]
<i>indexsort</i>	Index sort flag	Yes [Yes; No]
<i>locklist</i>	Maximum storage for lock lists	Database Server with local clients; 25 [4 - 60000] Database Server with local and remote clients: 50 [4 - 60000]
<i>locktimeout</i>	Lock timeout	-1 [-1;0 - 30000]
<i>logbufsz</i>	Log Buffer Size	8 [4 - 128]
<i>logfilsiz</i>	Size of log files	250 [4 - 4095]
<i>logprimary</i>	Number of primary log files	3 [2 - 128]
<i>logretain</i>	Log retain enable	OFF [ON,OFF]
<i>logsecond</i>	Number of secondary log files	2 [0 - 126]
<i>maxappls</i>	Maximum number of active applications	Database Server with local clients; 10 [1 - 1500] Database Server with local and remote clients: 20 [1 - 1500]
<i>maxfilop</i>	Maximum number of database files open per application	64 [2 - 32768]
<i>maxlocks</i>	Maximum percent of lock list before escalation	22 [1 - 100]

Table 23. Database manager configuration parameters whose defaults or ranges are influenced by the operating system on which the database manager was installed (continued)

Parameter	Description	Default [Range]
<i>mincommit</i>	Number of commits to group	1 [1 - 25]
<i>newlogpath</i>	Change the database log path	Null [any valid path]
<i>num_freqvalues</i>	Number of frequent values retained	10 [0 - 32767]
<i>num_iocleaners</i>	Number of asynchronous page cleaners	1 [0 - 255]
<i>num_ioservers</i>	Number of I/O servers	3 [1 - 255]
<i>num_quantiles</i>	Number of quantiles for columns	20 [0 - 32767]
<i>pckcachesz</i>	Package cache size	36 [1 - <i>applheapsz</i>]
<i>rec_his_retentn</i>	Recovery history retention period	366 [-1; 0 - 30000]
<i>seqdetect</i>	Sequential detection flag	Yes [Yes; No]
<i>softmax</i>	Log records to write before soft checkpoint	100 [1 - 100]
<i>sortheap</i>	Sort heap size	256 [16 - 524288]
<i>stat_heap_sz</i>	Statistics heap size	4384 [1096 - 524288]
<i>stmtheap</i>	Statement heap size	2048 [128 - 60000]
<i>userexit</i>	User exit enable	OFF [ON, OFF]
<i>util_heap_sz</i>	Utility heap size	5000 [16 - 524288]

See the *DB2 Administration Guide* for a description of the database configuration parameters.

Appendix C. Worksheets for Use with DataJoiner DDCS

This appendix contains the following worksheets for use when planning the installation of DDCS on your IBM software server.

- “MVS, VSE, and VM Server Worksheet”
- “OS/400 Server Worksheet” on page 263
- “Windows NT Configuration Worksheet” on page 264
- “Windows NT Connection to Server Worksheet” on page 264

For information about how to use the worksheets, see “Chapter 19. Configuring Communications for DRDA Data Sources” on page 117.

MVS, VSE, and VM Server Worksheet

Item	Parameter	Example	Your value	Note
1	NETID	SPIFNET		
2	Controller address	400009451902		
3	MODEENT	IBMRDB		
4	SSCP name	NYX		
5	APPL	NYM2DB2		
6	LU	NYX1GW01		1
7	PU	NYX1		1
8	IDBLK and IDNUM	05D 27509		1
9	Remote transaction program	(default)		2
10	Database location/name	NEW_YORK		

1. The PU, IDBLK, and IDNUM (items 7 and 8) must be the same for all DRDA server systems to which DataJoiner DDCS is connected. The LU can also be the same.
2. Appropriate values for item 9 are as follows:
 - For Microsoft SNA Server, X'07F6C4C2' for a connection to DB2 for MVS or DB2 for OS/400. For Communications Server, 076DB. Communications Server does not accept hexadecimal values.
 - The name of the remote database for connections to DB2 for VM or VSE. The default value is RDB_NAME.

For the DDCS administrator:

1. If you have already defined a connection to another DRDA server, write the LU, PU, and IDBLK and IDNUM values that you used in the table above.
2. If this is your first DRDA server connection, look at the **Server Properties** panel and write the control point name for your machine beside PU in the table above.

For the VTAM administrator:

DDCS gives applications on remote systems (such as Windows NT) access to data in your DB2 system. In order to make this possible, you must define DDCS to VTAM as follows:

1. Create an independent LU and a switched major node for the DDCS system. Use the PU value that the DDCS administrator wrote in the table above. If the LU, IDBLK, and IDNUM values are already written in the table, use these values also.
2. If the database is not already set up for remote communication in VTAM, set it up.
3. Write the following information for the DDCS administrator in the table:
 - The name of the network (NETID)
 - The locally-administered address or universal address of the communication controller
 - The entry in the mode table (MODEENT) that is used for communication. We recommend that you use mode IBMRDB, which is described in the *DRDA Connectivity Guide*.
 - The SSCP name
 - For the database, the APPL value
 - For the independent LU that corresponds to the DDCS system, the LU name
 - For the switched major node that corresponds to the DDCS system, the IDBLK and IDNUM values.
4. Give the worksheet to the database administrator.

For the database administrator:

DDCS gives applications on remote systems access to data in your DB2 system. In order to make this possible, you must perform the following steps:

1. Set up the database for remote communication. (For MVS, this includes adding entries to the SYSIBM.SYSLUNAMES and SYSIBM.SYSUSERNAMES tables). As an LU name, use the value that is written for “LU” in the table.
2. If you are not using the default TP name, write the value that you are using in the table beside remote transaction program. For MVS systems, the default (defined by DRDA) is X'07'6DB. For VSE and VM systems, the default is the RDB_NAME.

3. For MVS, write the LOCATION value in the last line of the table. For VSE or VM, write the RDB_NAME value in the last line of the table.
4. Give the worksheet to the DDCS administrator.

OS/400 Server Worksheet

Item	Parameter	Example	Your value	Note
1	Local network name	SPIFNET		1
2	Local adapter address	400009451902		2
3	Mode name	IBMRDB		3
4	Local control point	SYD2101A		4
5	Remote transaction program	(default)		5
6	Relational database name	NEW_YORK		6

For the AS/400 administrator:

DataJoiner DDCS gives applications on remote systems (such as Windows NT) access to data in your DB2 for OS/400 system. In order to make this possible, the DDCS administrator needs the following information:

1. The local network name. You can get this information by entering DSPNETA.
2. The local adapter address. You can get this information by entering WRKLIND (*TRLAN).
3. The mode name. You can get a list of mode names by entering WRKMODD. If the mode IBMRDB has been defined on your OS/400 system, we recommend that you use it.
4. The local control point name. You can get this information by entering DSPNETA.
5. The remote transaction program name. The default (defined by DRDA) is X'07'6DB.
6. The relational database name. You can get this information by entering DSPRDBDIRE.

Windows NT Configuration Worksheet

Item	Parameter	Example	Your value	Note
1	Network name	SPIFNET		1
2	Local control point name	NYX1		2
3	Local node ID	05D 27509		3
4	Mode name	IBMRDB		4
5	Local LU name	NYX1GW01		5
6	Local LU alias	NYX1GW01		

1. The Network name matches item 1 on the server worksheet.
2. For MVS, VSE, or VM, the Local control point name matches item 7 (PU) on the server worksheet. For OS/400, any unique node name can be used; in most cases, this value has already been set for you.
3. For MVS, VSE, or VM, the Local node ID matches item 8 (IDBLK and IDNUM) on the server worksheet. For OS/400, you can leave this blank.
4. The Mode name matches item 3 on the server worksheet.
5. The Local LU name matches item 6 (LU) on the MVS, VSE, and VM server worksheet.

Windows NT Connection to Server Worksheet

Item	Parameter	Example	Your value	Note
7	Link name	LINKHOST		
8	LAN destination address	400009451902		1
9	Symbolic destination name	DB2CPIC		
10	Control point name	NYX		2
11	Partner LU name	NYM2DB2		3
12	TP name	(default)		4
13	Target database name	NEW_YORK		5

1. The LAN destination address matches item 2 on the server worksheet.
2. The Control point name matches item 4 on the server worksheet.
3. For MVS, VSE, or VM, the Partner LU name matches item 5 (APPL) on the server worksheet. For OS/400, you can use the control point name.
4. The TP name is copied from the server worksheet. Appropriate values for item 12 are as follows:
 - X'07F6C4C2' for a connection to DB2 for MVS or DB2 for OS/400
 - The name of the remote database for connections to DB2 for VM or VSE. The default value is RDB_NAME.
5. The Target database name matches the last item on the server worksheet.

Appendix D. National Language Support

This appendix describes code page support and how character and code page conversion occur.

Replication restriction:

The replication administration tool does not support double-byte character strings (DBCS).

Conversion Between Different Code Pages

A code page maps each character from a character set, such as the Latin alphabet, to a numeric representation. Each code page is identified by a numeric identifier. For example, code page 850 represents the character “A” as hexadecimal 41.

Ideally, for optimal performance, your applications should always use the same code page as your database. However, this is not always practical or possible. DataJoiner provides support for character conversion that allows your application and database to use different code pages. Characters from one code page must be mapped to the other code page in order to maintain the meaning of the data.

When Does Character Conversion Occur?

Character conversion can occur in the following situations:

- When a client or application that accesses a database is running in a code page that is different from the code page of the database.

Most character data conversion for DataJoiner databases takes place on the database server machine by using internal code page mapping tables for single-byte code pages. For example, a Windows client application running in the Latin-1 code page 1252 can access a Windows NT database with character data that is encoded in the Latin-1 code page 1252.

You can minimize or eliminate client/server character conversion in some situations. For example, you could create a DataJoiner for Windows NT database that uses code page 850 to match an OS/2 client application environment that predominately uses code page 850.

- When a client or application that is importing a PC/IXF file runs in a code page that is different from the file being imported.

This data conversion will occur on the database client machine before the client accesses the database server. Additional data conversion may take

place if the application is running in a code page that is different from the code page of the database (as stated in the previous point).

Data conversion, if any, also depends on how the import utility was called. For more information, see the *DB2 Administration Guide*.

- When DDCS for Windows NT is used to access data on a DRDA server. In this case character conversion occurs by the receiver of data, as defined by the DRDA rules. For example, data that is sent to DB2 for MVS is converted to the appropriate MVS coded character set identifier (CCSID) by DB2 for MVS. The data that is sent back to the DDCS machine from DB2 for MVS is converted by DDCS. For more information, see the *DDCS Installation and Configuration Guide* for your platform.

Character conversion will **not** occur for:

- File names. You should either use the ASCII invariant set for file names or provide the file name in the hexadecimal values that are physically stored in the file system.
- Data that is targeted for or comes from a column assigned the FOR BIT DATA attribute, or data used in an SQL operation whose result is FOR BIT or BLOB data. In these cases, the data is treated as a byte stream, and no conversion occurs.¹¹ See the *DB2 SQL Reference* for unequal code page rules for assigning, comparing, and combining strings.
- Access to a DB2 for OS/2 Version 1.0 or Version 1.2 database server.

Deriving Code Page Values

The application code page is derived from the active environment when the database connection is made. The database code page is derived from the value that is specified, either explicitly or by default, at the time the database is created. The following defines how the *active environment* is determined in different operating environments, for example:

UNIX	In UNIX-based environments, the active environment is determined from the locale environment variables, which include information about language, territory, and code set.
OS/2	In OS/2, primary and secondary code pages are specified in the CONFIG.SYS file. You can use the chcp command to display and dynamically change code pages within a given session. See your OS/2 documentation for details.
DOS	In DOS, the active code page is determined by the value that is specified in the COUNTRY command in the CONFIG.SYS file.

11. However, a literal that is inserted into a column defined as FOR BIT DATA could be converted if that literal was part of an SQL statement which was converted.

You can use the `chcp` command to display and dynamically change code pages within a given session. See your DOS documentation for details.

Macintosh For the Macintosh operating system, if the `DB2CODEPAGE` environment variable is set, its value is taken as the application code page. If `DB2CODEPAGE` is not set, the Macintosh code page is derived from the Regional version code from the installed script. Supported Macintosh code pages include:

1275 (Apple Latin 1)	1282 (Apple Latin 2)
1280 (Apple Greek)	1283 (Apple Cyrillic)
1281 (Apple Turkish)	932 (Japanese)

Windows For Windows, if the `DB2CODEPAGE` environment keyword is set, its value is taken as the application code page. If `DB2CODEPAGE` is not set, the Windows code page is derived from the country ID, as specified in the `iCountry` value in the `[intl]` section of the Windows `WIN.INI` file. Supported Windows code pages include:

943 (Japanese)	1252 (Windows Latin 1)
949 (Korean)	1253 (Windows Greek)
950 (Traditional Chinese)	1254 (Windows Turkish)
1004 (Latin 1)	1255 (Windows Hebrew)
1250 (Windows Latin 2)	1256 (Windows Arabic)
1251 (Windows Cyrillic)	1381 (Simplified Chinese)

Windows 95 For Windows 95, if the `DB2CODEPAGE` environment keyword is set, its value is taken as the application code page. If `DB2CODEPAGE` is not set, the Windows 95 code page is derived from the ANSI code page setting in the Registry. Supported Windows 95 code pages include:

943 (Japanese)	1252 (Windows Latin 1)
949 (Korean)	1253 (Windows Greek)
950 (Traditional Chinese)	1254 (Windows Turkish)
1004 (Latin 1)	1255 (Windows Hebrew)
1250 (Windows Latin 2)	1256 (Windows Arabic)
1251 (Windows Cyrillic)	1381 (Simplified Chinese)

Windows NT For Windows NT, if the `DB2CODEPAGE` environment keyword is set, its value is taken as the application code page. If `DB2CODEPAGE` is not set, the Windows NT code page is derived from the ANSI code page setting in the Registry. Supported Windows NT code pages include:

943 (Japanese)	1252 (Windows Latin 1)
----------------	------------------------

949 (Korean)	1253 (Windows Greek)
950 (Traditional Chinese)	1254 (Windows Turkish)
1004 (Latin 1)	1255 (Windows Hebrew)
1250 (Windows Latin 2)	1256 (Windows Arabic)
1251 (Windows Cyrillic)	1381 (Simplified Chinese)

See Table 25 on page 271 for a complete list of supported locales, code page equivalents.

Supported Character Conversions

When data conversion occurs, conversion will take place from a source code page to a target code page.

The source code page is determined from the source of the data; data from the application has a source code page equal to the application code page, and data from the database has a source code page equal to the database code page.

The determination of target code page is more involved. Where the data is to be placed, as well as rules for intermediate operations, is considered:

- If the data is moved directly from an application into a database, with no intervening operations, the target code page is the database code page.
- If the data is being imported into a database from a PC/IXF file, there are two character conversion steps:
 1. From the PC/IXF file code page (source code page) to the application code page (target code page)
 2. From the application code page (source code page) to the database code page (target code page).

Exercise caution in situations where two conversion steps might occur. Make sure that you follow the supported character conversions that are listed in Table 24 to avoid a possible loss of character data.

- If the data is derived from operations that are performed on character data, where the source may be any of the application code page, the database code page, FOR BIT DATA, or for BLOB data, data conversion is based on a set of rules. Some or all of the data items may have to be converted to an intermediate result, before the final target code page can be determined. See the *DB2 SQL Reference* for a summary of these rules and for specific application with individual operators and predicates.

Table 24 on page 270 shows the code page conversions that are supported. Any code page can be converted to any other code page that is listed in the same row of the table. For example, code page 437 can be converted to 819, 850, 863, 1004, 1051, 1252, or 1257.

Character string consideration:

Character string conversions between multi-byte code pages, for example DBCS and EUC, may result in either an increase or a decrease in the length of the string.

Table 24. Supported Code Page Conversions

Code Pages	Countries
437, 819, 850, 863, 1004, 1051, 1252 ^d	Austria, Australia, Belgium, Canada, Denmark, Finland, France, Germany, Iceland, Italy, Latin America, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, UK, USA
813, 869, 1253 ^d	Greece
852, 912, 1250 ^d	Croatia, Czech Republic, Hungary, Poland, Romania, Serbia/Montenegro (Latin), Slovakia, Slovenia
855, 915, 1251 ^d	Bulgaria, FYR Macedonia, Serbia/Montenegro (Cyrillic)
857, 920, 1254 ^d	Turkey
862, 916, 1255 ^d	Israel
864, 1046, 1089, 1256 ^d	Arabic countries
866, 915, 1251 ^d	Russia
932, 942 ^a	Japan
943 ^a	Japan
938, 948 ^b , 950, 964	Taiwan
934, 944 ^{ce} , 949, 970 ^f	Korea
1381, 1383 ^c	People's Republic of China

Note:

- a** Code pages 932 and 942 are treated as equivalent by DataJoiner, so no character conversion is performed between them.
- b** Code pages 938 and 948 are treated as equivalent by DataJoiner, so no character conversion is performed between them.
- c** Code page 934 and 944 are treated as equivalent by DataJoiner, so no character conversion is performed between them.
- d** Code pages 1250, 1251, 1252, 1253, 1254, 1255, and 1256 are treated as server code pages by the Windows NT database manager.
- e** Code pages 934/944 to other Korean code pages will not be supported as these code pages are in the process of being phased out of IBM products. IBM Korea requires new support to be based on 949 and 970, which are both based on the KD5601 Korean standard.
- f** Code pages 949 and 970, and code pages 1381 and 1383 are identical except that they support different numbers of User Defined Characters. Some User Defined Characters will not be converted based on the character conversions provided by the operating system APIs.

Country Code and Code Page Support

Table 25 shows the languages and code sets that are supported by DataJoiner for Windows NT and how these values are mapped to country code and code page values that are used by the database manager.

The following list explains each column of the table:

Territory

Shows the territory associated with the supported code set.

Code Set

Shows the code set associated with the supported language. The code set is mapped to the DataJoiner code page.

Code Page

Shows the IBM-defined code page as mapped from the operating system code set.

Country Code

Shows the country code that is used by the database manager internally for providing country-specific support.

Country / Language

Lists the country associated with the country code.

Table 25. Supported Territories, Country Codes, Code Sets, and Code Pages

Territory	Code Set	Code Page	Country Code	Country / Language
AA	IBM-864	864	785	Arabic Countries
AR	IBM-864	864	785	Arabic Countries
AR	1256	1256	785	Arabic Countries
Aus	IBM-437	437	061	Australia (English)
Aus	IBM-850	850	061	Australia (English)
Aus	1252	1252	061	Australia (English)
BE	IBM-437	437	032	Belgium
BE	IBM-850	850	032	Belgium
BE	1252	1252	032	Belgium
BG	IBM-855	855	359	Bulgaria
BG	1251	1251	359	Bulgaria
CA	IBM-437	437	001	Canada (English)
CA	IBM-850	850	001	Canada (English)
CA	IBM-863	863	002	Canada (French)

Table 25. Supported Territories, Country Codes, Code Sets, and Code Pages (continued)

Territory	Code Set	Code Page	Country Code	Country / Language
CA	1252	1252	002	Canada (English)
HR	IBM-852	852	385	Croatia
HR	1250	1250	385	Croatia
CS	IBM-852	852	421	Czech Republic
CS	1250	1250	421	Czech Republic
CZ	IBM-852	852	421	Czech Republic
CZ	1250	1250	421	Czech Republic
DK	IBM-850	850	045	Denmark
DK	1252	1252	045	Denmark
FI	IBM-437	437	358	Finland
FI	IBM-850	850	358	Finland
FI	1252	1252	358	Finland
MK	IBM-855	855	389	FYR Macedonia
MK	1251	1251	389	FYR Macedonia
FR	IBM-437	437	033	France
FR	IBM-850	850	033	France
FR	1251	1251	033	France
DE	IBM-437	437	049	Germany
DE	IBM-850	850	049	Germany
DE	1252	1252	049	Germany
GR	ISO8859-7	813	030	Greece
GR	IBM-869	869	030	Greece
GR	1253	1253	030	Greece
HU	IBM-852	852	036	Hungary
HU	1250	1250	036	Hungary
IS	IBM-850	850	354	Iceland
IS	1252	1252	354	Iceland
IL	IBM-862	862	972	Israel
IL	1255	1255	972	Israel
IT	IBM-437	437	039	Italy
IT	IBM-850	850	039	Italy
IT	1252	1252	039	Italy

Table 25. Supported Territories, Country Codes, Code Sets, and Code Pages (continued)

Territory	Code Set	Code Page	Country Code	Country / Language
JP	IBM-932	932	081	Japan
JP	IBM-943	943	081	Japan
KR	IBM-949	949	082	Korea
Lat	IBM-437	437	003	Latin America
Lat	IBM-850	850	003	Latin America
Lat	1252	1252	003	Latin America
NL	IBM-437	437	031	Netherlands (Dutch)
NL	IBM-850	850	031	Netherlands (Dutch)
NL	1252	1252	031	Netherlands (Dutch)
NO	IBM-437	437	047	Norway
NO	IBM-850	850	047	Norway
NO	1252	1252	047	Norway
CN	IBM-1381	1381	086	PRC (Simp. Chinese)
PL	IBM-852	852	048	Poland
PL	1250	1250	048	Poland
PT	IBM-437	437	351	Portugal
PT	IBM-850	850	351	Portugal
PT	1252	1252	351	Portugal
RO	IBM-852	852	040	Romania
RO	1250	1250	040	Romania
RU	IBM-866	866	007	Russia
RU	1251	1251	007	Russia
SP	IBM-850	850	381	Serbia/Montenegro (Cyrillic)
SP	IBM-852	852	381	Serbia/Montenegro (Latin)
SP	1251	1251	381	Serbia/Montenegro (Cyrillic)
SP	1250	1250	381	Serbia/Montenegro (Latin)
YU	IBM-855	855	381	Serbia/Montenegro (Cyrillic)
YU	IBM-852	852	381	Serbia/Montenegro (Latin)
YU	1251	1251	381	Serbia/Montenegro (Cyrillic)
YU	1250	1250	381	Serbia/Montenegro (Latin)
SK	IBM-852	852	422	Slovakia
SK	1250	1250	422	Slovakia

Table 25. Supported Territories, Country Codes, Code Sets, and Code Pages (continued)

Territory	Code Set	Code Page	Country Code	Country / Language
SI	IBM-852	852	386	Slovenia
SI	1250	1250	386	Slovenia
ES	IBM-437	437	034	Spain
ES	IBM-850	850	034	Spain
ES	1252	1252	034	Spain
SE	IBM-437	437	046	Sweden
SE	IBM-850	850	046	Sweden
SE	1252	1252	046	Sweden
CH	IBM-437	437	041	Switzerland
CH	IBM-850	850	041	Switzerland
CH	1252	1252	041	Switzerland
TW	big5	950	088	Taiwan (T-Chinese)
TR	IBM-857	857	090	Turkey
TR	1254	1254	090	Turkey
GB	IBM-437	437	044	U.K. (English)
GB	IBM-850	850	044	U.K. (English)
GB	1252	1252	044	U.K. (English)
US	IBM-437	437	001	U.S.A. (English)
US	IBM-850	850	001	U.S.A. (English)
US	1252	1252	001	U.S.A. (English)

Table Notes:

1. DataJoiner uses the AIX environment locale name to determine the territory. The AIX environment defaults to the C (POSIX**) locale when one is not specified, the territory defaults to USA, and the code set defaults to IBM-850. These defaults are also applicable to countries that do not have locales defined yet, such as Australia and the Latin American countries. These defaults will be used until these countries have their own locales. The C locale is 7-bit ASCII-based, and therefore is a proper subset of locales that are based on IBM-850 and ISO8859-1. The national language support that is provided by DataJoiner for the U.S.A. is based on the IBM-850 and ISO8859-1, which are the same code sets that are used to support Australia and the Latin American countries. As a result, the default value is acceptable for Australia and the Latin American countries. Other countries must use their own locales to obtain proper national language support.

If you are in Australia or Latin America and are using the default locale, consider setting your locale to `En_US` (IBM-850) or `en_US` (ISO8859-1) before installing DataJoiner. This ensures that your code set requirements are matched with the language of the messages that you selected during installation. If you are in Latin America, you could also choose to set your locale to `Es_ES` (IBM-850) or `es_ES` (ISO8859-1) if you will be using Spanish as the language for the messages. It is also possible to use a Spanish locale and have English as the language for the messages. The default date format that is provided by DataJoiner will be the one for the United States or Spain, depending on the locale that is being used.

2. The Latin-1 Windows code page is 1004, which is a superset of code page 819 (ISO8859-1). DataJoiner performs round-trip conversion of character data that is encoded on code page 1004. However, for purposes of other NLS functions such as composing SQL identifiers, DataJoiner treats code page 1004 as equivalent to code page 819.
3. Canada uses CCSID 037 for DB2 for VSE or VM and DB2 for MVS, and CCSID 500 for OS/400 databases.
4. Neither AIX nor X/Open has registered territories to use with the Hebrew language (as identified by locale `iw`). Future additions of territory for this locale name will not impact DataJoiner NLS function.
5. DataJoiner does not support code page 1089 because code page 1089 does not encode all the characters of the IBM-defined EBCDIC Arabic code page 420. Converting character data that is encoded on code page 420 to code page 1089 would result in some characters being lost. DataJoiner supports code page 1046 since code page 1046 encodes all the characters of the EBCDIC Arabic code page 420.

DataJoiner does not support client/server connectivity between code page 1089 and code page 1046. As a result, Arabic language AIX database clients must use the same code page as the database that is defined on the server. Arabic language OS/2 and DOS database clients that run on code page 864 can connect to a DataJoiner server that is running in either 1089 or 1046.

6. Although the SBCS code page 850 is available in Japan, DataJoiner does not support this code page for a DBCS environment.
7. IBM Thailand provides its own sorting and collating routines to work with DataJoiner. Thai users should contact their local country support for additional information.

Date and Time Formats

The character string representation of date and time formats is the default format of the datetime values that are associated with the application's country code. This default format may be overridden by specification of the *F* format option when the program is precompiled or bound to the database.

The following is a description of the input and output formats for date and time:

- Input Time Format
 - There is no default input time format
 - All time formats are allowed as input for all country codes.
- Output Time Format
 - The default output time format is equal to the local time format.
- Input Date Format
 - There is no default input date format
 - Where the local format for date conflicts with an ISO, JIS, EUR, or USA date format, the local format is recognized for date input. For example, see the UK entry in Table 26.
- Output Date Format
 - The default output date format is equal to the local date format.

Table 26 shows a listing of the string formats for the various country codes.

Table 26. Date and Time Formats by Country Code

Country Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
785 Arabic	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 Australia (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
061 Australia	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
032 Belgium	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
055 Brazil	dd.mm.yy	JIS	LOC	LOC, USA, EUR, ISO
359 Bulgaria	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
001 Canada	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
002 Canada (French)	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
385 Croatia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 Czech Republic	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
045 Denmark	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO

Table 26. Date and Time Formats by Country Code (continued)

Country Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
358 Finland	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
389 FYR Macedonia	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
033 France	dd/mm/yyyy	JIS	EUR	LOC, EUR, ISO
049 Germany	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
030 Greece	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
036 Hungary	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
354 Iceland	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
972 Israel	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
039 Italy	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
081 Japan	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
082 Korea	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
001 Latin America (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
003 Latin America	dd-mm-yyyy	JIS	LOC	LOC, EUR, ISO
031 Netherlands	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
047 Norway	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
048 Poland	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
351 Portugal	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
086 PRC	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
040 Romania	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
007 Russia	dd/mm/yyyy	ISO	LOC	LOC, EUR, ISO
381 Serbia/Montenegro	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 Slovakia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
386 Slovenia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
034 Spain	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
046 Sweden	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
041 Switzerland	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
088 Taiwan	mm-dd-yyyy	JIS	ISO	LOC, USA, EUR, ISO
066 Thailand (2)	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
090 Turkey	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
044 UK	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO

Table 26. Date and Time Formats by Country Code (continued)

Country Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
001 USA	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
<p>Note:</p> <p>(1) Countries using the default C locale are assigned country code 001.</p> <p>(2) yyyy is in Buddhist era: Gregorian + 543 years.</p>				

Appendix E. Diagnosing Installation Problems on Windows 95/Windows NT

If you encounter installation problems, note the following:

- On Windows 95 and Windows NT, the Setup program attempts to replace certain files in the Windows system directory if the current versions of these files are lower than the versions that are required by DataJoiner. However, if a file is currently in use by another application or service, Setup will be unable to replace it. Setup logs the error message returned by the operating system during the failed copy operation and notifies you that an error has occurred. To remedy this situation:
 1. Shut down any other running applications.
 2. Stop any services that are running.
 3. Run Setup again.
- The system keeps a log (DB2.LOG) that tracks installation activities and uninstall activities and records information on any errors encountered during product installation or uninstall. The log is stored in the DB2LOG directory. If this directory exists, the Setup program uses it. If not, the Setup program creates the directory on the drive that contains the Windows directory. If space constraints or other errors prevent the log from being opened or written to, the installation or uninstall continues.

Handling Installation Errors

If an error occurs while you are running the Setup program, you should first read the error message text to find the probable cause and possible solution for the problem.

The following table lists installation errors that you may encounter and the action you should take to resolve the problem.

Table 27. Handling Installation Errors

Error	Action
Insufficient authority to run the Setup program or the user name is not valid.	<ol style="list-style-type: none">1. Logon with a user name that has administrator authority on the local machine and is no more than eight characters in length.2. Re-install the product.

Table 27. Handling Installation Errors (continued)

Error	Action
<p>The file <filename> has not been copied. Refer to the log <log_filename> for details. Install will continue.</p>	<p>The system file <filename> could not be copied to the Windows system directory. To recover:</p> <ol style="list-style-type: none"> 1. Shut down any other running applications. 2. Stop any services that are running. 3. Run the Setup program again.
<p>Windows NT program groups or items cannot be created.</p>	<p>To create a new program group, do the following:</p> <ol style="list-style-type: none"> 1. Create a common program group: <ol style="list-style-type: none"> a. Go to the Program Manager File item. b. Select New. c. Select Common Program Group. d. Select OK. e. Specify the properties of the group to be made. 2. Create a program item in that group: <ol style="list-style-type: none"> a. Go to the Program Manager File item. b. Select New. c. Select Program Item. d. Specify the properties of the new item to be made. <p>You do not need to re-install the product.</p>
<p>DB2 security service cannot be registered.</p>	<p>To register the DB2 security service, run DB2REGSC, specifying I for install, as follows:</p> <pre data-bbox="840 1225 1249 1251"><installation_path>\BIN\DB2REGSC I</pre> <p>You do not need to re-install the product.</p>

Table 27. Handling Installation Errors (continued)

Error	Action
<p>Environment variables cannot be added or updated.</p>	<p>In most cases, the Setup program will try to recover from this error and continue processing. The error message indicates if the install program continues or terminates.</p> <ul style="list-style-type: none"> • If it continues, you may need to correct some values using the System program group within the Program Manager Control Panel. You do not need to re-install the product. • If it does not continue, you need to correct the cause of the error before re-installing.
<p>Process environment variables may not be set.</p>	<p>This error may cause other install steps to fail. The error message gives the name of the environment variable that needs to be updated.</p> <p>If the Setup program gives errors, update the process environment variable using the SET command, then uninstall and re-install the product.</p>
<p>Default instance cannot be created, or the service for default instance cannot be registered.</p>	<p>If the default instance, DB2, could not be created, run DB2ICRT to create it, as follows</p> <pre style="margin-left: 40px;"><installation_path>\BIN\DB2ICRT DB2</pre> <p>You do not need to re-install the product.</p>

Table 27. Handling Installation Errors (continued)

Error	Action
Key cannot be added to the registry.	<p>In most cases, the Setup program will try to recover from this error and continue processing. The error message includes the following information:</p> <ul style="list-style-type: none">• The key, subkey, and value names.• The operation that was executing at the time of error, for example, OPEN, QUERY, or SET.• The return value from the operation.• Whether the installation process continues or terminates. <p>If the Setup program continues after an error, you do not need to re-install, but you may need to correct the key in the registry. If it does not continue, you may need to correct the cause of the error before re-installing.</p>

Appendix F. Resolving Problems Encountered by Applications That Predate Version 2.1.1

This appendix explains how to resolve problems that arise when certain applications, such as those based on DataJoiner Version 1.2, try to perform operations that are no longer valid in Version 2.1.1, to query or modify catalog tables that were updated for Versions 2.1 and 2.1.1, or to query catalog views that were updated for Version 2.1.1.

The word *applications* here refers to a wide range of programs and instructions; for example:

- Application program code
- Third-party utilities
- Interactive SQL queries
- Commands
- API invocation

This appendix does not describe:

- DataJoiner operations that are less likely to generate an error in Version 2.1.1 than in Version 1.2. These operations can have only a positive impact on existing applications.
- Inter-version differences that are common to DataJoiner and DB2. For a discussion of problems that can result from them, see “Appendix I. Incompatibilities between Releases”, in the *DB2 SQL Reference for common servers*.

The problems that this appendix addresses are those that can arise when applications that predate DataJoiner Version 2.1.1 try to:

- Query DataJoiner Version 2.1.1 catalog tables, or query DB2 for CS catalog views that have been updated for DataJoiner Version 2.1.1
- Modify DataJoiner Version 2.1.1 catalog tables

Querying System Catalog Tables and Views

This section explains:

- How DataJoiner catalog tables and DB2 for CS catalog views have been updated to support DataJoiner Version 2.1.1
- What problems can result when certain applications, such as those based on DataJoiner Version 1.2, try to query these tables or views
- How to resolve these problems

Changes

Changes have been made to several DataJoiner system catalog tables, and to certain DB2 for CS catalog views that support DataJoiner. This section discusses:

- Changes that could cause problems for applications designed to access catalog tables that were used by DataJoiner Version 1.2
- Changes that could cause problems for applications designed to access DB2 for CS views that have been updated to support the Spatial Extender.

Changes in Tables Used by DataJoiner Version 1.2

DataJoiner Version 1.2 uses three DB2 for CS catalog tables—SYSCOLUMNS, SYSINDEXES, and SYSTABLES—and two tables specific to DataJoiner—SYSREMOTEUSERS and SYSSERVERS. The following changes, listed by table, were made for DataJoiner Version 2.1 and retained in Version 2.1.1:

The SYSCOLUMNS Table: The following changes, listed by column, were made to this table:

HIGH2KEY	Non-character values are now in printable format rather than binary format.
LOW2KEY	Non-character values are now in printable format rather than binary format.
NULLS	The value D (not null with default) has been changed to N (not nullable).
REMOTE_TYPE	In Version 1.2, values denoted data types of columns of data source tables that DataJoiner referenced by nickname. In Version 2.1.1, these values are stored in REMOTE_TYPENAME.

The SYSINDEXES Table: In Version 1.2, the value in the CLUSTERRATIO column of this table was -1 if statistics were not gathered. In Version 2.1.1, the value is -1 either if statistics are not gathered or if detailed index statistics are gathered. In the latter case, an appropriate value is added to the CLUSTERFACTOR column.

The SYSREMOTEUSERS Table: The data type for this table's AUTHID column was changed from CHAR to VARCHAR.

The SYSSERVERS Table: The following changes, listed by column, were made to this table:

COLSEQ	Deleted from SYSSERVERS. In Version 2.1.1,
--------	--------------------------------------------

	this server option is denoted by a value (colseq) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
CONNECTSTRING	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (connectstring) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
CPURATIO	Data type changed from DOUBLE to FLOAT.
DATEFORMAT	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (DATEFORMAT) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
FOLDID	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (fold_id) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
IORATIO	Data type changed from DOUBLE to FLOAT.
PASSWORD	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (password) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
TIMEFORMAT	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (TIMEFORMAT) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.
TIMESTAMPFORMAT	Deleted from SYSSERVERS. In Version 2.1.1, this server option is denoted by a value (TIMESTAMPFORMAT) in the OPTION column of the SYSCAT.SERVER_OPTIONS catalog view.

The SYSTABLES Table: The following changes, listed by column, were made to this table:

PACKED_DESC	Data type changed from LONGVARCHAR to BLOB.
REL_DESC	Data type changed from LONGVARCHAR to BLOB.
VIEW_DESC	Data type changed from LONGVARCHAR to BLOB.

Changes in DB2 for CS Views That Support the Spatial Extender

The following DB2 for CS catalog views were changed to support the Spatial Extender, an optional facility that became available with DataJoiner Version 2.1.1. For information about the Spatial Extender, see *IBM DB2 Spatial Extender Administration Guide and Reference*.

The SYSCAT.DATATYPES View: The following columns were added to this view: EXTRA_LENGTH, TYPE_PRECEDENCE, and INSTANTIABLE.

The SYSCAT.FUNCPARMS View: The following columns were added to this view: PARMNAME, TYPE_PRESERVING, and MUTATED.

The SYSCAT.FUNCTIONS View: The following columns were added to this view: CONTAINS_SQL, DBINFO, RESULT_COLS, BODY, EFFECT, TYPE_PRESERVING, FUNC_PATH, and SELECTIVITY.

The SYSCAT.TRIGDEP View: A column named DTYPE was added to SYSCAT.TRIGDEP.

Problems

A variety of problems could occur. For example:

- If a DataJoiner Version 1.2 application does a qualified search on a column that takes a different value than it did before (for example, a search on NULLS in SYSIBM.SYSCOLUMNS for a value of D), the application might react differently than expected.
- If a DataJoiner Version 1.2 application queries a column whose data type has changed (for example, CPURATIO in SYSIBM.SYSSERVERS), too much or too little data might be returned.
- If a DB2 for CS application uses star notation (SELECT *) to query a view with new columns that the application doesn't recognize (for example, SYSCAT.DATATYPES, which has several new columns to support the Spatial Extender), the application will receive an error.

Resolution

Review the changes listed above to decide whether they affect your applications and, if so, what corrective action to take (for example, updating the application). So that any problems in accessing or maintaining catalog tables can be avoided, we strongly recommend that instead of querying these tables, you query the catalog views derived from them.

If you need a rough approximation of the degree of clustering, select both CLUSTERRATIO and CLUSTERFACTOR in the SYSCAT.INDEXES catalog view and choose the greater of the two values that you retrieve.

Modifying System Catalog Tables

This section explains:

- How the method for modifying system catalog tables changed in Version 2.1.1
- What problems can result when Version 1.2 applications try to modify Version 2.1.1 catalog tables
- How to resolve these problems

Change

For DataJoiner to perform operations on a specific data source, DataJoiner must associate an identifier (specifically, a server name) with that data source. In Version 1.2, you could create such an association by inserting appropriate values into the table `SYSIBM.SYSSERVERS`. You could also modify an association by updating `SYSIBM.SYSSERVERS`, and terminate an association by deleting a server name from `SYSIBM.SYSSERVERS`. In Versions 2.1 and 2.1.1, you use DDL to perform these same operations indirectly. Specifically, you create DataJoiner-to-data source associations with the `CREATE SERVER MAPPING` statement, modify them with the `ALTER SERVER MAPPING` statement, and terminate them with the `DROP` statement. These statements operate on `SYSCAT.SERVERS`, a catalog view derived from `SYSIBM.SYSSERVERS`. The changes that you make to the view are propagated to `SYSIBM.SYSSERVERS`.

For a user to access data sources from DataJoiner, DataJoiner must associate the ID under which the user connects to DataJoiner with the IDs under which the user connects to these data sources. In Version 1.2, you could create such an association by inserting appropriate values into the table `SYSIBM.SYSREMOTEUSERS`. You could also modify an association by updating `SYSIBM.SYSREMOTEUSERS`, and terminate an association by deleting an ID from `SYSIBM.SYSREMOTEUSERS`. In Versions 2.1 and 2.1.1, you use DDL to perform these same operations indirectly. Specifically, you create associations between IDs with the `CREATE USER MAPPING` statement, modify them with the `ALTER USER MAPPING` statement, and terminate them with the `DROP` statement. These statements operate on `SYSCAT.REMOTEUSERS`, a catalog view derived from `SYSIBM.SYSREMOTEUSERS`. The changes that you make to the view are propagated to `SYSIBM.SYSREMOTEUSERS`.

Problem

If you issue an `INSERT`, `UPDATE`, or `DELETE` statement against `SYSIBM.SYSSERVERS`, `SYSIBM.SYSREMOTEUSERS`, or any of DataJoiner's other system catalog tables, the statement will fail.

Resolution

To modify SYSIBM.SYSSERVERS or SYSIBM.SYSREMOTEUSERS, use the SERVER MAPPING or USER MAPPING DDLs, as described in “Change” on page 287.

Appendix G. Where to Learn More about DataJoiner, DB2 for CS, and Replication Products

This appendix lists IBM books about DataJoiner, DB2 for CS, and Replication Administration; states how to obtain these books; and tells you where to go on the Internet to learn more about DataJoiner.

DataJoiner, DB2 for CS, and Replication Publications

Table 28 on page 290 lists the DataJoiner, DB2 for CS, and Replication books applicable to installing, configuring, administrating, using, and running applications against DataJoiner. The *DataJoiner for UNIX Systems Planning, Installation, and Configuration Guide* and the *DataJoiner for Windows NT Systems Planning, Installation, and Configuration Guide* are provided in hardcopy with DataJoiner. In addition, these two books and all other DataJoiner books are provided in softcopy formats (PostScript, HTML, and PDF) on the product CD-ROM. All other books in Table 28 on page 290 are provided in PostScript; most are also provided in HTML (the two exceptions are the DB2 for CS Software Developer Kit publications). Additionally, most of the DB2 for CS books are provided in INF format (see Table 28 on page 290).

To understand how the DataJoiner books in Table 28 on page 290 are organized, it is important to understand how DataJoiner and DB2 for CS are interrelated. DataJoiner provides a “superset” of DB2 for CS. The two products share common functions and syntax; therefore, information that is common to DataJoiner and DB2 for CS is documented in the DB2 for CS books. The DataJoiner books listed in Table 28 on page 290 document the function and syntax that DataJoiner has *in addition to* the function and syntax that it shares with DB2 for CS.

Table 28 on page 290 does not list all of the DB2 for CS books. View or print a DB2 for CS book to see the publications list for all DB2 for CS books.

If you order Classic Connect, you will receive additional documentation:

- The *DataJoiner Classic Connect Installation, Configuration, and Reference Guide*, V2.1.1
- The *DataJoiner Classic Connect Data Mapper Installation and User's Guide*, V2.1.1
- A program directory

Table 28. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner

Book Name	Form Number	File Prefix	INF
DataJoiner Version 2.1.1 Books			
<i>DataJoiner for Windows NT Systems Planning, Installation, and Configuration Guide</i>	SC26-9150	DJXN3	no
This book covers capacity planning, resource management, installation, and configuration tasks for IBM DataJoiner on Microsoft Windows NT operating systems.			
<i>DataJoiner for UNIX Systems Planning, Installation, and Configuration Guide</i>	SC26-9145	DJXG7	no
This book covers capacity planning, resource management, installation, and configuration tasks for IBM DataJoiner on AIX and Solaris operating systems.			
<i>DataJoiner Administration Supplement</i>	SC26-9146	DJXD5	no
This book provides information that assists DBAs and other system administrators of DataJoiner with performing administrative tasks. It includes a product overview section, security considerations, data source identification steps, database utility notes, performance considerations, database system monitor reference data, large object information, and explain tool examples.			
<i>DataJoiner Application Programming and SQL Reference Supplement</i>	SC26-9148	DJXK5	no
This book provides SQL statements, descriptions of system catalog data, guidelines, and other information for application programmers. With this information, application programmers can use DataJoiner to perform multiple tasks in a distributed database environment—tasks such as creating nicknames by which to reference tables and views, invoking functions and stored procedures, passing SQL directly to databases for processing, and using server options to optimize query performance.			
<i>DataJoiner Generic Access API Reference</i>	SC26-9147	DJXM4	no
This book explains how to create a generic access module that allows you to use existing drivers or to create new drivers to gain access to an unlimited set of data sources.			
<i>DataJoiner Classic Connect Installation, Configuration, and Reference Guide</i>	SC26-9319	DJXC5	no

Table 28. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

Book Name	Form Number	File Prefix	INF
<p>This book provides information about the DataJoiner Classic Connect for MVS product. The audience for this information includes application programmers, database administrators, network administrators, system administrators, and system programmers. The book documents key tasks required to set up Classic Connect including: planning your setup; installing components using SMP/E; configuring the AIX and Windows NT client, data servers, enterprise servers, and network communications; managing instances; and creating relational data maps for IMS and VSAM data. This book also contains all system messages that pertain to Classic Connect.</p>			
<i>DataJoiner Classic Connect Data Mapper Installation and User's Guide</i>	SC26-9318	DJXZ3	no
<p>This book provides information about the DataJoiner Classic Connect data mapper utility for Windows. The audience for this information includes system programmers, database administrators, or anyone that needs to produce relational equivalent structures for IMS and VSAM data. The book documents key tasks required to set up and use the data mapper in the Windows environment: installing product files, starting the product, and generating meta data grammar for input to the DataJoiner Classic Connect meta data utility.</p>			
<i>DataJoiner Messages and Problem Determination Guide</i>	SC26-9149	DJXP4	no
<p>This book describes the messages and codes issued by DataJoiner and Classic Connect instances. For messages that report errors, the book explains the cause of the errors and recommends corrective actions. The book also provides guidelines on using diagnostic tools to isolate and understand problems.</p>			
<i>DB2 Spatial Extender Administration Guide and Reference</i>	SC26-9316	DJXS1	no
<p>This book provides instructions for spatially enabling a DataJoiner database, an introduction to spatial capabilities using geometry data types and functions, descriptions of spatial data exchange formats, an SQL and message reference for spatial data, and appendices containing the standard representations of spatial reference systems.</p>			
DB2 for CS and Replication Books			
<i>DB2 Information and Concepts Guide</i>	S20H-4664	SQLG0	no
<p>Provides product and conceptual information to anyone who needs a comprehensive overview of the DB2 products. It is useful when deciding which DB2 products suit your environment. It also includes a glossary of terms used in the book.</p>			

Table 28. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

Book Name	Form Number	File Prefix	INF
<i>DB2 Administration Guide</i>	S20H-4580	SQLD0	yes
Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.			
<i>DB2 Database System Monitor Guide and Reference</i>	S20H-4871	SQLF0	yes
Includes a description of how to use the Database System Monitor and a description of all the data elements for which information can be collected.			
<i>DB2 Command Reference</i>	S20H-4645	SQLN0	yes
Provides the reference information needed to use system commands and the DB2 command line processor to execute database administrative functions. Describes the commands that can be entered at an operating system command prompt or in a shell script to access the database manager. Explains how to invoke and use the command line processor, and describes the command line processor options. Provides a description of all the database manager commands.			
<i>DB2 API Reference</i>	S20H-4984	SQLB0	yes
Provides information about the use of application programming interfaces (APIs) to execute database administrative functions. Presents a description of APIs and the data structures used when calling APIs, as well as detailed information on the use of database manager API calls in applications written in the supported programming languages.			
<i>DB2 SQL Reference</i>	S20H-4665	SQLS0	yes
Is intended to serve as a reference for syntax and rules governing the use of SQL statements. Syntax diagrams, semantic descriptions, rules, and examples are provided for the SQL statements. Catalog views, product maximums, release-to-release incompatibilities, and a glossary are also included in this book.			
<i>DB2 Application Programming Guide</i>	S20H-4643	SQLA0	yes
Discusses the application development process and how to code, compile, and execute application programs that use embedded SQL to access the database. It includes discussions on programming techniques and performance considerations for the application programmer.			
<i>DB2 Call Level Interface Guide and Reference</i>	S20H-4644	SQLL0	yes

Table 28. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

Book Name	Form Number	File Prefix	INF
Is a guide and reference manual for programmers using the Call Level Interface. DB2 Call Level Interface is a callable SQL interface based on the X/Open CLI specification and is compatible with Microsoft Corporation's ODBC.			
<i>DB2 Messages Reference</i>	S20H-4808	SQLM0	yes
Lists messages and explanations. Each explanation includes the action to be taken when a message or code is issued.			
<i>DB2 Problem Determination Guide</i>	S20H-4779	SQLP0	yes
Provides information that helps in determining the source of errors, recovering from problems, and describing and reporting defects.			
<i>DDCS User's Guide</i>	S20H-4793	SQLC0	yes
Provides concepts, programming guidelines, and general information about the DDCS products.			
<i>DB2 Replication Guide and Reference</i>	S95H-0999	DB3E0	no
Describes how to plan, configure, administer, and operate IBM replication products, including the Apply and Capture programs.			
DB2 for CS Platform-Specific Books			
<i>DB2 SDK for AIX Building Your Applications</i>	S20H-4780	SQLA3	yes
This book provides environment setup information and step-by-step instructions to compile and link DB2 applications on the AIX operating system.			
<i>DB2 SDK for Windows 95 and NT Building Your Applications</i>	S33H-0310	SQLA6	yes
This book provides environment setup information and step-by-step instructions to compile and link DB2 applications on Windows 95 and NT operating systems.			
<i>DB2 SDK for Solaris Building Your Applications</i>	S34H-0890	SQLA5	yes

Table 28. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

Book Name	Form Number	File Prefix	INF
This book provides environment setup information and step-by-step instructions to compile and link DB2 applications on Sun's Solaris operating system.			

How to Order, View, and Print Publications

Use order number SBOF-5289 to request one hardcopy of each of the DataJoiner, DB2 for CS, and Replication books shown in Table 28 on page 290.

To view online documentation, follow the instructions located in the README files on the CD-ROM. Most of the books in Table 28 on page 290 are provided as HTML files and can be viewed with an HTML browser. You can also view INF versions of many DB2 for CS books. Instructions for installing the INF reader on AIX are provided in the DB2 README files; on NT operating systems, the INF reader is installed automatically. DataJoiner and Replication information is not provided in INF format.

To print individual books, follow the instructions provided in the README files on the CD-ROM. PostScript files for all the books are provided.

Internet Resources

The following Internet resources provide additional information about DataJoiner.

World Wide Web

The following DataJoiner-specific Web site contains general and technical (frequently asked questions) product information. The address of the site is:

<http://www.ibm.com/software/data/datajoiner>

Also available online are the most current versions of books in the DB2 library. You can view books in the DB2 library by clicking the Library link from the following address:

<http://www.ibm.com/software/data/pubs/techinfo.html>

Internet Newsgroups

DataJoiner questions, answers, and discussions can be found in:

- bit.listserv.db2-l
- comp.databases
- comp.databases.ibm-db2

Appendix H. DataJoiner Classes and Services

This appendix describes:

- Classes you can take to learn about DataJoiner
- Services to help you plan to use DataJoiner, and to install and configure it

DataJoiner Classes

IBM offers classes that teach you how to install, use, and maintain DataJoiner. These classes are described in this section.

For more information, or to enroll in any IBM class, call 1-800-IBM-TEACH (1-800-426-8322) and refer to the IBM US Course Code. For locations outside the United States, contact your IBM representative.

Class descriptions will also be maintained at the DataJoiner Web site. The DataJoiner URL is:

<http://www.ibm.com/software/data/datajoiner>

Using DataJoiner

IBM US Course Code DW202

Duration

2 days

Format

Lecture with classroom exercises.

This course introduces the student to DataJoiner and its powerful multidatabase server capabilities. After completing this course, students should be able to effectively use DataJoiner to perform simple and complex distributed requests. They should also be able to monitor and tune SQL queries, accounting for the capabilities and characteristics of diverse DataJoiner data sources. Areas covered include:

- Global optimization
- Multi-vendor query considerations
- Nicknames
- Basic security
- An introduction to the DataJoiner catalog
- DataJoiner query performance
- The DataJoiner Explain tool

- The DataJoiner Database System Monitor

Who Should Take This Course

This course is appropriate for anyone who will be using, managing, installing, or maintaining a DataJoiner multiple database environment.

Prerequisite

SQL experience. You can obtain this experience by attending the “SQL Workshop,” IBM US Course Code CF120.

DataJoiner Administration

IBM US Course Code DW212

Duration

3 days

Format

Lecture with classroom exercises.

This course trains the student to install, configure, and manage a secure DataJoiner multidatabase server environment. Areas covered include:

- Installing DataJoiner
- Generating and managing the DataJoiner database
- Configuring DataJoiner
- Enabling DataJoiner client access to remote data sources
- DataJoiner security
- DataJoiner server performance

Who Should Take This Course

This course is appropriate for anyone who will be managing, installing, or maintaining a DataJoiner multiple database environment.

Prerequisite

DataJoiner knowledge or experience. You can obtain this experience by attending “Using DataJoiner,” IBM US Course Code DW202.

DataJoiner Services

IBM provides services for DataJoiner that include assistance with planning, installing, and configuring the product. The assistance is customized to your individual environment and takes place in two phases.

First Phase: Planning

The first phase helps you plan the installation and configuration of DataJoiner, and to configure network systems so that DataJoiner can communicate optimally with all data sources and clients. This phase includes:

- Assessing general readiness

- Defining clients
- Defining data sources
- Assessing applications
- Defining backup and recovery strategies for DataJoiner
- Configuring DataJoiner database parameters
- Identifying test queries for system validation
- Defining security requirements

Second Phase: Implementation

The second phase focuses on implementing the plan developed in the planning phase. It includes:

- Installing DataJoiner
- Configuring data sources
- Providing access to data source tables and views
- Installing and configuring remote clients
- Validating and documenting the environment
- Providing final turnover to the customer

At the end of this phase, active remote and local clients can access multiple data sources through DataJoiner.

DataJoiner services can be combined with replication services if you are interested in replicating data across a heterogeneous database environment. For more information about DataJoiner and replication services, contact your IBM representative or see the DataJoiner Web page. The DataJoiner URL is:

<http://www.ibm.com/software/data/datajoiner>

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.

Licenses 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or

distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

ADSTAR	IIN
Advanced Peer-to-Peer Networking	IMS
AIX	IMS/ESA
APPN	Language Environment
AS/400	MVS
AT	MVS/ESA
CICS	MVS/XA
CICS/6000	NetView
Client Acces	Operating System/2
Current	Operating System/400
DATABASE 2	OS/2
DataGuide	OS/390
DataJoiner	OS/400
DataPropagator	RACF
DataRefresher	RETAIN
DB2	RISC System/6000
DFSMS	RS/6000
Distributed Relational Database Architecture	RT
DPROP	SP
DRDA	SQL/DS
Extended Services for OS/2	SQL/400
HACMP/6000	System/390
IBM	VisualAge
	VTAM

Intel is a registered trademark of the Intel Corporation in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Solaris is a trademark of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT[®], and the Windows logo are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Index

A

access

replication

configuring in DataJoiner for
Windows NT 212

accessing data

through DataJoiner DDCCS 10

through Net.Data 9

accessing data sources

configuration overview 51

DB2RA 67

Microsoft SQL Server through
ODBC 85

Oracle RDB through ODBC 89

administration

authorization requirements 203

overview 164

Administrator's Toolkit

Database Director 7

DB2 Performance Monitor 7

hardware requirements 24

overview 6

Visual Explain 7

after-image columns,

description 184

APPC

adding a connection

using Microsoft SNA Server
Manager 126

configuring a node

using Communications
Server 121

configuring a partner LU

using Communications
Server 122

configuring a side information profile

using Communications
Server 122

configuring a transaction

program name
using Communications
Server 123

configuring devices

using Communications
Server 121

configuring network connections

using Communications
Server 121

APPC (continued)

configuring the Mode

using Communications
Server 122

configuring using

Communications Server 121

configuring using Microsoft SNA Server Manager 124

defining server properties

using Microsoft SNA Server
Manager 124

installing the DLC

communications protocol

using Microsoft SNA Server
Manager 124

installing the link service

using Microsoft SNA Server
Manager 125

saving your configuration

using Microsoft SNA Server
Manager 135

setting a symbolic destination name

using Microsoft SNA Server
Manager 133

setting local LU properties

using Microsoft SNA Server
Manager 130

setting remote LU properties

using Microsoft SNA Server
Manager 132

setting the Mode

using Microsoft SNA Server
Manager 130

application development

using Net.Data 9

using Software Developer's
Kit 8

application requesters,

supported 252

application server

configuring DataJoiner as 117

security considerations when
using SNA 117

worksheets 261, 263

Apply for Windows NT and

Windows 95

authentication 224

binding the package 223

Apply program

Apply qualifier 173

authorization requirements 204

choosing location 176

control tables, use of 174

for UNIX platforms

invocation parameters 232

operating 230

scheduling 234

starting 231

stopping 234

instances of 173

introduction 172

location of 188

password file feature 173

planning considerations 28

processing cycle 176

push vs. pull configuration 188

refresh versus differential

refresh 180

relationship to other tables 173

replicating changes 172

run-time processing

statements 187

subsetting target columns and

rows 185

synchronization with Capture

triggers 183

trail tables 175

user ID 204

applying changes to target

tables 172

AS/400 access

configuring DDCCS 117

gathering server

information 118

overview 10

worksheet 263

ASN qualifier 241

ASNAPPLY command

for UNIX platforms 232

ASNSTOP command

for UNIX platforms 234

AT command

Apply program

for UNIX platforms 234

authentication

end-user for DB2 DPROF for

Microsoft Jet 215

- authentication (*continued*)
 - end-user for Windows NT and Windows 95 224
- authorization
 - Apply for Windows NT and Windows 95 224
 - coordinating 192
 - for administrator 203
 - requirements
 - Apply program 204
 - Capture program 203
 - for administration 203
 - planning 202
 - system-specific privilege levels 205
- automating replication
 - activities 161
- B**
- before-image columns
 - column value 185
 - description 184
 - NULL value 185
 - restrictions 193
 - usage 184
- binding
 - applications and utilities to database 77
 - Apply for Windows NT and Windows 95 223
 - Capture and Apply programs in DB2 222
 - DRDA AS options 246
 - replication source, target and control servers 220
 - Windows NT Capture and Apply programs 214
- binding packages 64
- book
 - conventions x
 - icons list x
- books, ordering and viewing 294
- BOOKSHELF environment
 - variable 147
- C**
- CAE
 - overview 27
- CALL procedures 187
- Capture and Apply programs
 - binding in DB2 222
 - running 230
- Capture program
 - authorization requirements 203
 - introduction 168
 - pruning CCD tables 181
- Capture program (*continued*)
 - staging data 181
 - user ID 203
- Capture triggers
 - capturing changes 169
 - control tables, used by 171
 - gaps in target data 192
 - introduction 168
 - relationships to other tables 169
 - synchronization with the Apply program 183
 - usage 169
- capturing changes, overview 169
- cataloging
 - database directory entry for DRDA 64
 - DCS directory entry 63
- CCD (consistent change data) tables
 - and the Capture triggers 169
 - description 171, 181
 - internal 183
 - naming schema 241
 - nicknames 242
 - putting at the source 221
 - qualifier 241
 - staging changed data 181
- CD (change data) tables
 - and consistent change data 181
 - pruning of 183
- CD-ROM
 - installing DataJoiner from 33
 - installing the replication administration tool from 217
- character conversion
 - supported code pages 268
 - when occurs 265
- Client Application Enabler
 - overview 27
- Client Pack
 - components included 3
 - DB2 Client Application Enabler, overview 3
- Client Pack, components
 - included 3
- clients 27
 - accessing DataJoiner 27
 - connecting the replication administration tool to DataJoiner 219
 - installation restriction 27
 - operating systems supported 4
 - replication
 - connecting in Windows NT 213
- CNTLSVR.REX 229
- code page values
 - how determined 266
- columns
 - before-image and after-image 184
 - fragmenting 185
 - subsetting 185
- combining data, techniques 186
- command line processor 7
- differences 15
- commands
 - ASNAAPPLY
 - for UNIX platforms 232
 - ASNSTOP
 - for UNIX platforms 234
 - AT
 - Apply for UNIX platforms 234
 - CREATE DATABASE 43
 - db2ckmig 39
 - db2start 42
 - db2stop 42
 - echo 147
 - MIGRATE DATABASE 38
 - NET START 42
 - NET STOP 42
 - RESTORE DATABASE 38
- communication protocols
 - software requirements 25
- Communications Server
 - configuring APPC for DRDA 121
- compilers supported 28
- components, selecting 34
- concepts, data replication 179
- condensed
 - CCD tables 183
 - staging tables 183
- configuring
 - APPC 117
 - data sources
 - DB2 through DB2RA 67
 - DRDA 57
 - Informix 81
 - Microsoft SQL Server through ODBC 85
 - Oracle RDB through ODBC 89
 - Oracle through Net8 91
 - Oracle through SQL*Net 91
 - overview 51
 - Sybase SQL Anywhere 103
 - Sybase through Open Client 97
 - Teradata 105

- configuring *(continued)*
 - through a generic data access module 113
 - DDCS 117
- connecting
 - DB2 databases in the replication administration tool 219
 - for replication
 - databases in DataJoiner for Windows NT 212
- connecting to the Internet using Net.Data 9
- connection affected by ODBC 220
- CONNECTSTRING option 100
- consistent change data (CCD) tables 169
- console window 228
- consolidate tables 191
- Control Center
 - managing the replication environment 164
- control server
 - choosing location 176
- Control server
 - description 155
- control tables
 - Apply trail 175
 - creating 165, 171, 221
 - customizing SQL, description of 166
 - generated by the replication administration tool 165
 - overview 164
 - pruning 171
 - register tables 171
 - subscription columns 175
 - subscription events 175
 - subscription set 174
 - subscription statements 175
 - subscription targets
 - members 174
 - used by Apply program 174
 - used by Capture triggers 171
- conventions
 - book x
- conversion
 - data types 235
- coordination of
 - target security and source security 192
- creating
 - databases 43
 - logical unions 187
 - sample database 35
 - user names 32
- creating and running generated SQL
 - order of activities associated with 166
- customizing
 - SQL 166
- D**
- data access modules
 - loading dynamically 52
 - overview 51
- data consistency, subscription sets 179
- data replication
 - administrating 163
 - concepts of 179
 - overview 157
 - tools used for 161
- data sources
 - access methods 51
 - DB2RA 67
 - DRDA 57
 - Informix 81
 - Microsoft SQL Server 85
 - Oracle Net8 91
 - Oracle RDB through ODBC 89
 - Oracle SQL*Net 91
 - Sybase Open Client 97
 - Sybase SQL Anywhere 103
 - Teradata 105
 - accessed by DataJoiner 26
 - configuration overview 51
- data type conversion
 - DB2 to Informix 236
 - DB2 to Microsoft Jet 238
 - DB2 to MS SQL Server 237
 - DB2 to Oracle 236
 - DB2 to SQL Anywhere 237
 - DB2 to Sybase 237
- database, creating 43
- database access, two-phase
 - commit 6
- database administration tools
 - DataJoiner for Solaris systems
 - restriction 6
 - overview 6
- Database Director
 - getting started 141
 - hardware requirements 24
 - overview 7
- database manager
 - starting methods 42
 - stopping methods 42
- DataJoiner
 - as a multivendor server 160
- DataJoiner *(continued)*
 - classes 295
 - client access to 27
 - components included
 - Database Director 7
 - DB2 Client Application Enabler 4
 - DB2 Performance Monitor 7
 - Net.Data 9
 - Visual Explain 7
 - data type conversions 235
 - differences between
 - versions 283
 - for Windows NT
 - connecting clients for replication 213
 - connecting databases for replication 212
 - creating databases 212
 - getting started 41
 - hardware requirements 24
 - installing from CD-ROM 33
 - nicknames, use of 167
 - overview 3, 167
 - platforms supported 4
 - programs
 - DB2 Client Application Enabler 3
 - Net.Data 3
 - Software Developer's Kit 3
 - software requirements 25
 - the replication administration tool as a client 219
 - user mappings 206
 - Windows NT differences 15
 - DataJoiner DDCS
 - configuration worksheets 261, 263, 264
 - overview 10
 - DataJoiner web site 294
 - DataPropagator NonRelational 157
 - DataPropagator Relational V1, support for DataJoiner 157
 - date
 - formats 276
 - DB2
 - log 169
 - DB2 Client Application Enabler
 - accessing databases 5
 - platforms supported 4
 - software requirements 25
 - DB2 log 169
 - DB2 Performance Monitor
 - overview 7
 - using 143

- DB2 replication
 - gaps in target data 192
 - DB2 web site 294
 - DB2ACCOUNT environment variable 148
 - DB2BQTIME environment variable 150
 - DB2BQTRY environment variable 150
 - DB2CHKPTR environment variable 149
 - db2ckmig command 39
 - DB2CODEPAGE environment variable 149
 - DB2COMM environment variable 151
 - DB2DBDFT environment variable 150
 - DB2DBMSADDR environment variable 149
 - DB2DMNBCKCTRL environment variable 149
 - DB2INSTANCE environment variable 147, 150
 - DB2INSTPROF environment variable 150
 - DB2IQTIME environment variable 151
 - DB2LOADREC environment variable 150
 - DB2LOG directory 32, 279
 - DB2NBADAPTERS environment variable 151
 - DB2NBCHECKUPTIME environment variable 152
 - DB2NBINTRLISTENS environment variable 151
 - DB2NBRECVBUFFSIZE environment variable 152
 - DB2NBRECVNCBS environment variable 152
 - DB2NBSENDNCBS environment variable 152
 - DB2NBSESSIONS environment variable 151
 - DB2OPTIONS environment variable 151
 - DB2PATH environment variable 147
 - DB2RA data sources
 - accessing 67
 - accessing using nicknames and pass-through 67 (continued)
 - updating the node directory 73
 - DB2RA data sources (continued)
 - accessing using nicknames and pass-through 67 (continued)
 - updating the node directory 73
 - configuring communications
 - APPC 118
 - IPX/SPX 70
 - TCP/IP 68
 - outside the heterogeneous environment 76
 - binding packages used by DataJoiner 77
 - cataloging a database directory entry 77
 - updating the node directory 77
 - DB2RQTIME environment variable 151
 - DB2SERVICETPINSTANCE environment variable 152
 - DBCS, not supported for replication 194
 - DDCS
 - configuring using SNA 118
 - configuring using TCP/IP 135
 - using for connectivity 163
 - developing applications
 - using Net.Data 9
 - using Software Developer's Kit 8
 - differential refresh 155, 180
 - directory names, language identifiers for 32
 - distributed database connection services (DDCS) 163
 - distributed-unit-of-work access 6
 - DIXCOMM environment variable 52, 148
 - double-byte character string, not supported for replication 194
 - DPROP NR (DataPropagator NonRelational) 157
 - DRDA
 - application requesters, supported 252
 - Application Server
 - bind options 246
 - CCSID support 251
 - configuring 251
 - DB2 for VM,
 - considerations 253
 - description 245
 - PTFs for DB2 for MVS 252
 - PTFs for DB2 for VM 252
 - DRDA (continued)
 - Application Server (continued)
 - PTFs for DDCS 252
 - PTFs for OS/400 253
 - supported ARs 252
 - supported functions 245
 - configuring SNA 118
 - configuring TCP/IP 135
 - DRDA data sources
 - accessing 57
 - using DataJoiner DDCCS 62
 - binding DataJoiner packages 64
 - cataloging a database directory entry 64
 - cataloging a DCS directory entry 63
 - configuring TCP/IP 118, 135
 - updating the node directory 62
 - using nicknames and pass-through 57
 - configuring SNA LU 6.2 118
 - configuring TCP/IP 135
 - updating catalog views 59
 - updating the node directory 58
 - DRDA server
 - worksheets 261, 263
- ## E
- editing the replication administration tool logic 229
 - editing the replication administration tool SQL 229
 - electronic information 294
 - environment variables
 - BOOKSHELF 147
 - DB2ACCOUNT 148
 - DB2BQTIME 150
 - DB2BQTRY 150
 - DB2CHKPTR 149
 - DB2CODEPAGE 149
 - DB2COMM 151
 - DB2DBDFT 150
 - DB2DBMSADDR 149
 - DB2DMNBCKCTRL 149
 - DB2INSTANCE 147, 150
 - DB2INSTPROF 150
 - DB2IQTIME 151
 - DB2LOADREC 150
 - DB2NBADAPTERS 151
 - DB2NBCHECKUPTIME 152
 - DB2NBINTRLISTENS 151
 - DB2NBRECVBUFFSIZE 152

environment variables (*continued*)
 DB2NBRECVNCBS 152
 DB2NBSSENDNCBS 152
 DB2NBSSESSIONS 151
 DB2OPTIONS 151
 DB2PATH 147
 DB2RQTIME 151
 DB2SERVICETPINSTANCE 152
 determining 147
 DJXCOMM 148
 HELP 147
 INCLUDE 147
 LIB 147
 PATH 147
 setting 147
 updating 281
 errors, installation 279

F

fixed disk
 hardware requirements 24
 fragmentation
 horizontal 186
 vertical 185
 fragmenting target tables 185
 full refresh
 description 155, 180

G

gaps in target data 192
 general library information 289
 generated SQL
 editing 229
 running 230
 generic data sources
 accessing through a generic data
 access module 113
 getting started, steps 41
 graphical user interface (GUI)
 replication administration tool
 creating control tables
 from 165
 replication administration tool's
 overview 163
 using for administration 163

H

hardware requirements
 DataJoiner 24
 fixed disk 24
 memory 24
 Net.Data 24
 Software Developer's Kit 24
 HELP environment variable 147
 heterogeneous server 155

horizontal subsetting
 description 186
 hosts file and Teradata 105

I

IBM replication solution
 restrictions 193
 tools 157
 icons
 description x
 INCLUDE environment
 variable 147
 Informix
 accessing 81
 esql/rt libraries 81
 NET START requirement 42
 setting DJXCOMM for 83
 Version 7 data access module 81
 installation
 errors 279
 log 31, 279
 planning 23
 pre-installation notes 31
 replication
 Apply for Windows NT 211
 DataJoiner for Windows
 NT 211
 replication administration
 tool 217
 selectable components 34
 selectable products 33
 steps 33
 verifying 34
 instance, creating default 281
 instances
 setup for replication
 DataJoiner for Windows
 NT 211
 internal CCD tables
 reducing network load 183
 Internet information 294
 introduction to the replication
 administration tool
 Apply program 172
 Capture program 168
 Capture trigger 168
 invocation parameters
 Apply program
 for UNIX platforms 232

J

joins
 description 192
 multitable 187
 of tables 191

L

languages, DataJoiner available
 in 32
 levels, privileges 205
 LIB environment variable 147
 library information, general 289
 local client
 Apply for Windows NT 211
 logic
 CNTLSVR.REX 229
 editing the replication
 administration tool logic 229
 SRCESVR.REX 229
 TARGSVR.REX 229
 TBLSPACE.REX 229
 logical partitioning key and
 replication 186
 logical unions, creating 187
 logs
 install 31, 279

M

managing databases using Database
 Director 7
 mapping user IDs 206
 memory
 estimating requirements 24
 Microsoft SNA server 119
 Microsoft SNA Server Manager
 configuring APPC for
 DRDA 124
 Microsoft SQL Server 99
 required client software 26
 MIGRATE DATABASE
 command 38
 migration
 commands 38
 correcting inhibitors 38
 databases 37
 instances 37
 restrictions 37
 selective 39
 monitoring databases
 using DB2 Performance
 Monitor 7
 using Performance Monitor 143
 monitoring performance
 using Windows NT Performance
 Monitor 143
 multisite union 192
 multitable joins and unions 187
 multivendor access 160
 MVS
 gathering DDOS
 information 119

MVS (*continued*)
gathering server
information 118
worksheet 261

N

national language support (NLS)
character conversion
overview 265
supported 268
client support 265
code page values
deriving 266
code pages
overview 265
supported 271
code sets
supported 271
country codes
supported 271
data and time formats 276
host CCSID 265
territories
supported 271
Net.Data
hardware requirements 24
overview 9
NET START command 42
NET STOP command 42
Net8 91
network
configuring OS/400 access 117
newsgroups for DataJoiner 294
node directory
updating for DRDA data sources
heterogeneous access 58
using DataJoiner DDCS 62
non-IBM database
CCD tables and Capture
triggers 241
nicknames in DataJoiner 242
non-IBM server
description 155
non-IBM database
control tables 241
pruning control table 242
register control table 242
table names and qualifiers 241
notices 299

O

ODBC
accessing Microsoft SQL Server
data sources 85
accessing Oracle RDB data
sources 89

ODBC (*continued*)
accessing Sybase SQL
Anywhere 103
running applications 47
ODBC affecting preferences 220
Open Client 97
operating
Apply program
for UNIX platforms 230
Oracle
accessing RDB through
ODBC 89
accessing through Net8 91
accessing through SQL*Net 91
required client software 27
tnsnames.ora file 91
ordering publications 294
OS/400 access
configuring DDCS 117
gathering server
information 118
worksheet 263

P

packages
binding 64
partitioning key and replication 186
password file
Apply for Windows NT and
Windows 95 224
authorization requirements 205
using 173
passwords
coordination of 192
PATH environment variable 147
Performance Monitor
differences 15
getting started 143
hardware requirements 24
planning
authorization requirements 202
replication considerations 28
point-in-time target tables 179
preferences affected by ODBC 220
Preferences notebook, setting the
replication administration
tool 219
preparing the replication
environment for DB2 DPROP for
Microsoft Jet 214
primary key
errors when updating 186
point-in-time target table 179
user copy target table 179
printing publications 294
privilege levels 205
products, selecting 33
products supported
as sources 161
as targets 161
promote function 192
pruning
CCD tables 181
CD and UOW tables 183
control tables 171
pruning control table 242
publications 294
push and pull configuration
choosing a configuration 191
comparison of 188
description of 188

R

refresh, the Apply program 168,
172
refresh copying 180
register control table 242
register tables 171
replica 179
replica target tables 191
replication
Apply program 172
Capture and Apply program
requirements 202
Capture program, introduction
to 168
Capture triggers, introduction
to 168
connecting all databases 217
description 156
differential refresh 155
full refresh 155
hardware requirements 198
objects, description of 165
overview 157, 163
overview of components 14
performance tip for 183
planning considerations 28
replication administration
tool 161
set up in the replication
administration tool 217
setting up
DataJoiner for Windows
NT 211
software requirements 198
sources 165
subscription sets 165
supported environments 158,
159, 160

- replication (*continued*)
 - supported sources and targets 197
 - tables 165
 - transaction based 183
 - update anywhere 191
 - description 156
- replication administration tool
 - Apply program 172
 - creating control tables 164
 - creating control tables from 165
 - customizing replication requests 166
 - hardware requirements 198
 - introduction 161
- replication targets, sets 156
- requirements
 - hardware 23
 - software 25
- RESTORE DATABASE
 - command 38
- restrictions
 - before-image columns 193
 - IBM replication solution 193
- rows, subsetting 186
- run-time processing, SQL statements 187
- running applications
 - database client 47
- running Capture and Apply programs 230
- running the replication administration tool SQL 230
- S**
- sample database
 - creating 35
- scheduling
 - Apply program for UNIX platforms 234
- security 202
 - description 192
 - domain database access 13
 - local database access 13
 - overview 13
 - remote database access 13
 - trusted clients 14
- security service
 - registering 280
- sequence for the replication administration tool SQL 228
- servers
 - binding from the replication administration tool 220
 - control 155
 - non-IBM 155
- servers (*continued*)
 - source 156
 - target 156
- Service Control Manager, Apply for Windows NT 225
- SET SERVER OPTIONS statement
 - CONNECTSTRING option 100
- setting environment variables 147
- setting up
 - replication
 - DataJoiner for Windows NT 211
 - setup
 - finalizing Apply for Windows NT 223
- Setup program 31
- SNA
 - adding a connection
 - using Microsoft SNA Server Manager 126
 - configuring a node
 - using Communications Server 121
 - configuring a partner LU
 - using Communications Server 122
 - configuring a side information profile
 - using Communications Server 122
 - configuring a transaction program name
 - using Communications Server 123
 - configuring devices
 - using Communications Server 121
 - configuring for DDCS 118
 - configuring the Mode
 - using Communications Server 122
 - configuring using
 - Communications Server 121
 - configuring using Microsoft SNA Server Manager 124
 - defining server properties
 - using Microsoft SNA Server Manager 124
 - gathering DDCS information 119
 - gathering server connection information 120
 - gathering server information 118
- SNA (*continued*)
 - installing the DLC communications protocol
 - using Microsoft SNA Server Manager 124
 - installing the link service
 - using Microsoft SNA Server Manager 125
 - saving your configuration
 - using Microsoft SNA Server Manager 135
 - setting a symbolic destination name
 - using Microsoft SNA Server Manager 133
 - setting local LU properties
 - using Microsoft SNA Server Manager 130
 - setting remote LU properties
 - using Microsoft SNA Server Manager 132
 - setting the Mode
 - using Microsoft SNA Server Manager 130
- SNA server 119
- Software Developer's Kit
 - hardware requirements 24
 - overview 8
- software key
 - updating 282
- software requirements
 - communication protocols 25
 - DataJoiner 25
 - DataJoiner data sources 26
 - DB2 Client Application Enabler 25
 - supported compilers 28
- solution, data replication 157
- source
 - control tables, on the 171
 - description 165
 - security, coordination with target 192
 - supported 161
 - table
 - definition of 163
 - views 191
- source server
 - description 156
- sources
 - description 156
- sources and targets
 - supported for replication 197
- SQL
 - before and after, description 187

- SQL (*continued*)
 - creating and running of
 - generated 166
 - editing the replication
 - administration tool SQL 229
 - generated
 - deferring the running of 166
 - description 166
 - order of creating and
 - running 166
 - running the replication
 - administration tool 230
 - statements, run-time processing
 - of 187
- SQL access plans
 - tuning 7
 - viewing using Visual Explain 7
- SQL/DS
 - gathering server
 - information 118
 - worksheet 261
- SQL*Net 91
- SQL Server (Microsoft) 99
- SRCESVR.REX 229
 - CCD table name 241
- staging data description 181
- staging table
 - complete 183
 - description of 181
- starting
 - Apply program
 - for UNIX platforms 231
- starting Capture and Apply
 - programs 230
- starting the replication
 - administration tool 227
- stopping
 - Apply program
 - for UNIX platforms 234
- subscription
 - columns tables description 175
 - events tables description 175
 - member
 - description 163, 179
 - sets
 - Apply qualifier 173
 - description 163, 165
 - description of 179
 - run-time processing
 - statements 187
 - tables, description of 174
 - target views 192
 - statements tables
 - description 175
 - subscription (*continued*)
 - targets members tables
 - description 174
 - subscription members
 - members description 156
 - subscription sets
 - description 156
 - subsetting
 - horizontal 186
 - source tables 185
 - vertical 185
 - supported sources and targets
 - for replication 197
 - Sybase
 - accessing SQL Anywhere 103
 - accessing through Open
 - Client 97
 - required client software 27
 - specifying a connect string 100
 - synchpoint 183
 - synchronization of
 - target security and source
 - security 192
 - system configuration
 - using DataJoiner 5
 - using DataJoiner DDCCS
 - Multi-User Gateway 11
 - using Net.Data 9
- T**
 - table space, specifying in logic 229
 - target
 - description 165
 - security, coordination with
 - source 192
 - supported 161
 - tables
 - CCD description 181
 - fragmenting of 185
 - point-in-time 179
 - pruning of 183
 - replica 179
 - subsetting columns in 185
 - subsetting rows in 186
 - types of 179
 - user copy 179
 - views 192
 - target server 156
 - TARGSVR.REX 229
 - TBLSPACE.REX 229
 - TCP/IP
 - with DB2RA 68
 - Teradata
 - accessing 105
 - required client software 27
 - the replication administration tool
 - authorization requirements 203
 - console window 228
 - creating control tables 221
 - initial startup 227
 - installing 217
 - planning considerations 28
 - setting preferences 219
 - software requirements 198
 - using 228
 - time
 - formats 276
 - tools
 - used for data replication 161
 - trademarks 301
 - triggers
 - See also* Capture triggers 221
 - trusted clients 14
 - two-phase commit
 - configuring 52
 - two-phase commit database
 - access 6
- U**
 - unions
 - multisite 192
 - multitable 187
 - unit-of-work (UOW) tables
 - and the Capture triggers 169
 - pruning of 183
 - update-anywhere replication
 - description 156, 191
 - user copy target tables 179
 - user IDs
 - mapping 206
 - specified in the replication
 - administration tool 220
 - user name
 - creating 32
 - naming rules 32
 - using the replication administration
 - tool 166, 228
- V**
 - verifying installation 34
 - vertical subsetting 185
 - viewing publications 294
 - views
 - definitions 191
 - registering as a source table 187
 - source, description 191
 - target, description 192
 - used as sources 191
 - using to perform joins 191
 - Visual Explain
 - getting started 141

Visual Explain (*continued*)
overview 7

VM

gathering DDCS
information 119
gathering server
information 118
worksheet 261

VSE

gathering DDCS
information 119
gathering server
information 118
worksheet 261

VTAM 119

W

wildcards 228

Windows NT

connecting to DB2 databases for
replication 219
updating the registry 147

worksheets

DataJoiner DDCS
configuration 261
hardware requirements 24

WWW information 294

X

XA libraries

link-editing for two-phase
commit 52

Readers' Comments — We'd Like to Hear from You

**DB2 DataJoiner[®] for Windows NT Systems
Planning, Installation,
and Configuration Guide**

Publication No. SC26-9150-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



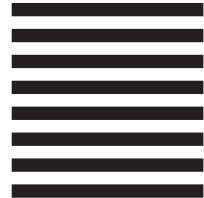
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM CORPORATION
Department BWE/H3
PO Box 49023
San Jose, CA 95161-9945



Fold and Tape

Please do not staple

Fold and Tape



Part Number: CT6DDIE

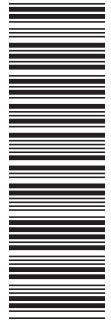


Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-9150-02



CT6DDIE



Spine information:



DB2 DataJoiner[®] for Windows
NT Systems

**Planning, Installation, and Configuration
Guide**

Version 2
Release 1 Modification
1