



DB2 DataJoiner[®] for AIX Systems

Planning, Installation, and Configuration Guide

Version 2 Release 1 Modification 1



DB2 DataJoiner[®] for AIX Systems

Planning, Installation, and Configuration Guide

Version 2 Release 1 Modification 1

Note

Before using this information and the product it supports, please read the general information under "Notices" on page 371.

Second Edition (July 1998)

This edition replaces and makes obsolete the previous edition, SC26-9145-00. The technical changes for this edition are summarized under "Chapter 2. What's New in DataJoiner Version 2?" on page 15, and are indicated by a vertical bar to the left of a change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, BWE/H3
P. O. Box 49023
San Jose, CA 95161-9023
U.S.A.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1995, 1998. All rights reserved.**

Copyright © 1993-1998 CROSS ACCESS Corporation. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xiii
Tables	xv
Welcome to DataJoiner	xvii
Highlighting Conventions	xvii
Graphical Conventions	xviii
How to Read the Syntax Diagrams	xviii
Road Map	xx

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
Heterogeneous Data Replication	13
Chapter 2. What's New in DataJoiner Version 2?	15

Part 2. Planning for and Installing DataJoiner 19

Chapter 3. Planning for Installation	21
Hardware Requirements	21
General Requirement Information	21
Memory and Disk Requirements	21
Software Requirements	25
Operating System Requirements	25
Client Access to DataJoiner	25
DRDA Databases	26
Requirements for DataJoiner Data Sources	27
Communication Protocol Requirements	28
Graphical User Interface Requirements	29
Transaction Managers	29
High Availability using HACMP	30
DB2 SNMP Subagent	30
Two-Phase Commit Requirements	30
Supported Compilers for DataJoiner	30

ADSTAR Distributed Storage Manager	31
Prerequisites for DataJoiner Options	31
DB2 Client Application Enabler (djx_02_01_01.client).	31
Command Line Processor Option (djx_02_01_01.clp)	32
SDK Options (djx_02_01_01.sdk...)	32
DataJoiner Executables Options (djx_02_01_01.db2).	32
Code Page Conversion Option (djx_02_01_01.conv)	32
Communication Support Options (djx_02_01_01.cs)	32
DDCS Options (djx_02_01_01.ddcs)	33
DB2 Database Director Option (djx_02_01_01.dd).	33
DB2 Visual Explain Option (djx_02_01_01.ve)	33
DB2 Spatial Extender Option (djx_02_01_01.db2se)	34
Chapter 4. Installing DataJoiner	35
Installation Planning.	35
Install Prerequisite Software	35
Understand DataJoiner Security.	35
Decide on Appropriate Authorization	35
Decide Whether You Need Additional Password Encryption	35
Decide Which Client Protocols to Use.	36
Replication Considerations	36
Migrating From Previous Releases	37
Other Considerations	42
A Common Installation Problem for AIX 3.2	44
Installation.	44
Step 1: Restore the Product Files to the Target Machine	45
Step 2: (Optional) Set Up Additional Password Encryption	47
Step 3: Link DataJoiner to Data Source Client Software	47
Step 4: Set up an Instance of the Product	53
Chapter 5. Setting up an Instance	55
Step 1: Create an Instance of the Product	55
Step 2: Set Environment Variables	57
Step 3: Set AIX Processes	60
Step 4: Start the Instance	61
Step 5: Create a DataJoiner Database	61
Notes:	61
Parameters:	61
Creating the Database	64
Step 6: Connect DataJoiner to Data Sources	64
Step 7: Connect Clients to DataJoiner.	64
Step 8: Enable a DataJoiner Database for Spatial Data	65
Part 3. Connecting DataJoiner to Data Sources	67
Chapter 6. Configuring Access to Data Sources	69
Link-Editing Data Access Modules	70
Loading Data Access Modules	70

Overview of Configuration Steps	71
Chapter 7. Accessing DRDA Data Sources	73
Accessing DRDA Data Sources using Nicknames and Pass-through.	73
Step 1: Configure Network Communications.	76
Configuring TCP/IP for DRDA3 Data Sources	76
Configuring SNA Server/6000 for DRDA Data Sources	78
Step 2: Update the Node Directory	92
Step 3: Update DataJoiner Catalog Views with Server and User Information	93
Step 4: Test the Connections to the Data Source	95
Accessing DRDA Data Sources using DataJoiner DDCS	95
Step 1: Configure Network Communications.	96
Step 2: Update the Node Directory	96
Step 3: Catalog a DCS Directory Entry	97
Step 4: Catalog a Database Directory Entry	97
Step 5: Bind Packages used by DataJoiner and Other Applications	97
Chapter 8. Accessing DB2RA Data Sources	101
Accessing DB2RA Data Sources using Nicknames and Pass-through	101
Step 1: Configure Network Communications.	101
Configuring TCP/IP for DB2RA Data Sources	101
Configuring IPX/SPX for DB2RA Data Sources.	104
Configuring APPC for DB2RA Data Sources.	107
Step 2: Update the Node Directory	125
Step 3: Update DataJoiner Catalog Views with Server and User Information	125
Step 4: Test the Connections to the Data Source	128
Accessing DB2RA Data Sources Outside the Heterogeneous Environment	128
Step 1: Configure Network Communications.	128
Step 2: Update the Node Directory	129
Step 3: Catalog a Database Directory Entry	129
Step 4: Bind Packages used by DataJoiner and Other Applications	129
Chapter 9. Accessing Informix Data Sources	131
Using the Informix-Net Version 5 Data Access Module	131
Using the Informix Version 7 Data Access Module.	137
Chapter 10. Accessing Microsoft SQL Server and Sybase Data Sources Using Sybase Open Client	143
Specifying a Connect String for Open Client.	146
Syntax	146
Examples	147
Chapter 11. Accessing Microsoft SQL Server through the Visigenic ODBC Driver	149
Chapter 12. Accessing Oracle Data Sources Using SQL*Net or Net8	155
Chapter 13. Accessing Data Sources through CrossAccess	159

Chapter 14. Accessing Data Sources Using a Generic Data Access Module	165
Chapter 15. Configuring for Two-Phase Commit	169
Enabling Data Sources for Two-Phase Commit	170
Configuring DataJoiner for Two-Phase Commit	171

Part 4. Connecting Clients to DataJoiner 173

Chapter 16. Accessing Local and Remote Clients	175
---	-----

Chapter 17. TCP/IP Support for Remote Clients	177
Configuring DataJoiner Instances for TCP/IP Clients	177
Configuring the Client	178
Tuning Network Options for Remote Failures Using TCP/IP	179
Configuration Examples	179
Configuring a DataJoiner Instance for TCP/IP Clients	182
Configuring CAE 2.1.1 for AIX or SDK 2.1.1 for an AIX TCP/IP Client	183
Configuring CAE 2.1.1 for OS/2 or SDK 2.1.1 for an OS/2 TCP/IP Client	185

Chapter 18. IPX/SPX Support for Remote Clients	187
Configuring IPX/SPX for DataJoiner	187
Updating the Database Manager Configuration File for IPX/SPX	188
File Server Addressing:	188
Direct Addressing:	189
Registering the DB2 Server	189
Setting DB2COMM for IPX/SPX	190

Chapter 19. APPC Support for Remote Clients	191
Configuring DataJoiner and AIX Clients	191
Configuring DataJoiner for CAE for AIX APPC Clients	194
Configuring a CAE for AIX APPC Client	197
Configuring DataJoiner and Configuring OS/2 Clients	199
Configuring a DB2 for OS/2 Version 2 Client	199

Chapter 20. Controlling Communication Support Initialization	205
---	-----

Chapter 21. Configuring the DataJoiner DRDA Application Server and Clients	207
APPC Support	207
Configuring DataJoiner as a DRDA AS	207
Step 1: Configure SNA Server/6000	207
Step 2: Update the Database Manager Configuration	212
Step 3: Set DB2COMM for APPC	212
Step 4: Stop All Connections	212
Step 5: Restart the Instance	212
Configuring Clients as DRDA ARs	213
DB2 for MVS DRDA AR Sample	213

Part 5. Maintaining DataJoiner 217

Chapter 22. Maintaining the DataJoiner Environment	219
DB2 and DDCS Coexistence with DataJoiner	219
Password Considerations for OS/2 Clients	220
Maintaining DataJoiner	220
Installing DataJoiner Program Temporary Fixes (PTFs)	220
Deleting Options on AIX 3.2	220
Deleting Options on AIX 4.1	221
Chapter 23. Maintaining Instances	223
Starting and Stopping a DataJoiner Instance	223
Using DataJoiner Commands	223
Using the kill Command	223
When to Stop and Restart Instances	223
Updating Instances	224
Starting a DataJoiner Instance at System Boot	226
Multiple Instances	226
Removing Instances	228
Authorizing Access within an Instance	229

Part 6. Replicating Data 231

Chapter 24. Data Replication - What is it?	233
The IBM Data Replication Solution	233
Support of Replication Environments	234
Replication Solution and Product Environment	234
Multivendor Data Sources and Targets	236
Chapter 25. The Replication Tools	237
The Replication Administration Tool	237
Administering Replication: Overview	239
How the Replication Administration Tool Manages the Replication Environment	240
DataJoiner	243
The Capture Program for DB2 Sources	243
Capture Triggers for Non-IBM Sources	244
Capture Triggers Overview	244
How the Capture Triggers Capture the Data Changes	244
Tables Used by the Capture Triggers	246
The Apply Program	247
The Apply Qualifier	248
Control Tables Used by the Apply Program	249
The Apply Program and Control Servers	251
The Apply Processing Cycle	251
Improving Apply Performance for Sybase or Microsoft SQL Server on AIX	251
Chapter 26. Data Replication Concepts	253
Subscription Members and Sets	253
Types of Copies	253
Full-Refresh and Differential Refresh	254

Changed Data Tables	254
CD and CCD Tables	255
Pruning the CD and CCD Tables	257
Transaction-Based versus Transaction-Consistent Replication: Using Internal CCD Tables to Reduce Network Load	257
Before-Image Columns and After-Image Columns	258
Before-Image Column Values	258
NULL Values in Before-Image Columns	258
Subsetting Target Tables	259
Vertical Subsetting	259
Horizontal Subsetting	259
Combining Data	260
SQL Statements and Stored Procedures	261
Pull versus Push Apply Design	261
Update-Anywhere Replication	264
Views for Sources and Targets	264
Source Views	264
Target Views	265
Preventing Gaps in the Target Table	265
Coordination of Source Security and Target Security	265
Promote Functions	265
IBM Replication Solution Restrictions	266
Chapter 27. Replication Requirements	269
Supported Sources and Targets	269
The Replication Administration Tool Hardware Requirements	270
DB2 DPROP for Microsoft Jet Hardware Requirements	270
The Replication Administration Tool Software Requirements	270
DB2 DPROP for Microsoft Jet Software Requirements	273
Capture and Apply Program Requirements	273
Authorization Requirements	274
Authorization Requirements for Administration	274
Authorization Requirements for the Capture Program	275
Authorization Requirements for the Apply Program	275
System-Specific Privilege Levels	276
Mapping of User IDs	277
Chapter 28. Setting up the DataJoiner Environment for Replication	283
Replication in DataJoiner for AIX	283
Installing DataJoiner	283
Setting up an Instance	284
Configuring Database Connections	284
Connecting Clients to DataJoiner	285
What to Do Next	285
Chapter 29. Installing the Replication Administration Tool and Connecting All Databases	287
Installing the Replication Administration Tool	287
Configuring Access from the Replication Administration Tool to DataJoiner and DB2	288

Setting DB2CODEPAGE for DataJoiner for AIX Access	289
Setting Administrative Preferences	289
Binding to Source, Target, and Control Servers.	290
Creating Replication Control Tables	290
Binding the Capture and Apply Programs in DB2 Systems	292
Finalizing the Setup for the Apply Program in AIX	292
Binding the Apply Program in AIX	292
Providing End-User Authentication in AIX.	293
Chapter 30. Starting and Using the Replication Administration Tool	295
Starting the Replication Administration Tool	295
General Steps for Setting up Replication	295
Editing the Replication Administration Tool Logic	296
Editing the Replication Administration Tool-Generated SQL	297
Running the Replication Administration Tool-Generated SQL	298
Running Capture and Apply	298
Chapter 31. DataJoiner with the Replication Administration Tool: Data Typing	299
DB2 to Oracle Replication.	300
DB2 to Informix	300
DB2 to Microsoft SQL Server, Sybase, or Sybase SQL Anywhere	301
DB2 to Microsoft Jet	302
Chapter 32. Non-IBM Databases: Table Names, Control Tables, and the CCD	305
Table Names and Qualifiers	305
Control Tables in a Non-IBM Database	305
CCD Table and Capture Triggers	305
CCD Nicknames in DataJoiner	306
The Register Control Table	306
Appendix A. DRDA Application Server	307
Supported DRDA Functions	307
Bind Options Supported by the DataJoiner DRDA Application Server	307
Restrictions	311
Configuration Considerations.	312
CCSID Support	312
Supported DRDA Application Requesters.	317
MVS DRDA Application Requesters	317
DDCS DRDA Application Requesters	317
DB2 for VM DRDA Application Requesters	317
DB2 for AS/400 DRDA Application Requesters	318
Appendix B. Configuration Parameters	319
Appendix C. National Language Support	325
Setting Up Your Language Environment	325
Setting Up Your Application and Database NLS Environments.	327
Deriving Code Page Values	328
Deriving Locales in Application Programs.	330

How DataJoiner Derives Locales	331
Programming Considerations.	331
Coding SQL Statements	332
Coding Remote Stored Procedures.	332
Precompiling and Binding	332
Executing an Application	332
A Note of Caution	332
Conversion Between Different Code Pages	333
When Does Character Conversion Occur?	333
Character Substitutions During Conversions.	334
Supported Character Conversions	335
Character Conversion Expansion Factor	336
Locale, Code Page, and Host CCSID Support	337
DBCS Code Page Support	341
EUC Code Page Support	342
DataJoiner Code Page Support	343
Specifying Code Pages	343
Character Sets	345
DBCS Character Sets	345
Extended UNIX Code (EUC) Character Sets	345
UCS-2 Character Sets	346
Character Set for Identifiers	347
Coding of SQL Statements	348
Collating Sequences	348
Datetime Values	348
Date and Time Formats	351

Appendix D. Resolving Problems Encountered by Applications That Predate

Version 2.1.1	355
Linking DataJoiner Libraries to Clients and Data Sources in AIX	355
Change	356
Problem.	356
Resolution	356
Starting and Stopping Classic Connect Instances	356
Change	356
Problem.	356
Resolution	356
Querying System Catalog Tables and Views.	357
Changes	357
Problems	359
Resolution	359
Modifying System Catalog Tables	360
Change	360
Problem.	360
Resolution	360

Appendix E. Where to Find Out More about DataJoiner, DB2 for CS, and Replication Products

DataJoiner, DB2 for CS, and Replication Publications	361
--	-----

How to Order, View, and Print Publications	365
Internet Resources	365
Appendix F. DataJoiner Classes and Services	367
DataJoiner Classes	367
Using DataJoiner.	367
DataJoiner Administration	368
DataJoiner Services.	368
First Phase: Planning	368
Second Phase: Implementation	369
Notices	371
Trademarks	373
Index	375
Readers' Comments — We'd Like to Hear from You	387

Figures

1. DataJoiner Server	5
2. Accessing Multiple DataJoiner Servers	6
3. Net.Data	10
4. DataJoiner DDCS	12
5. Example of output that results from running the djxlink.sh script	49
6. CREATE DATABASE Command Syntax	61
7. DataJoiner Application Requester Heterogeneous Access	74
8. DRDA Data Source Planning Template	75
9. DRDA Data Source through TCP/IP	77
10. Token-Ring SNA DLC Profile	80
11. Token-Ring Link Station Profile	81
12. Control Point Profile	82
13. LU 6.2 Local LU Profile	84
14. Partner LU 6.2 Location Profile	85
15. Mode Profile	87
16. LU 6.2 Side Information Profile	89
17. SNA System Defaults	91
18. DB2RA Data Source through TCP/IP	102
19. DB2RA Data Source Planning Template for TCP/IP	104
20. APPC Sample Values: Connecting DataJoiner to a DB2 for OS/2 V2 Data Source	108
21. APPC Planning Template: Connecting DataJoiner to a DB2 for OS/2 V2 Data Source	109
22. Token-Ring SNA DLC Profile	115
23. Link Station Profile Screen	116
24. Control Point Profile	117
25. LU 6.2 Local LU Profile	119
26. Partner LU 6.2 Location Profile	120
27. Mode Profile	121
28. LU 6.2 Side Information Profile	123
29. SNA System Defaults	124
30. Informix V5 Data Access Module	132
31. Informix V7 Data Access Module	137
32. Sybase Data Access Module	144
33. Visigenic ODBC Data Access Module	151
34. ORACLE Data Access Module	157
35. System Catalog Tables	163
36. Remote TCP/IP Clients	177
37. TCP/IP Planning Template	181
38. Configuring a DataJoiner Instance for TCP/IP Clients	182
39. CAE 2.1.1 for AIX or SDK 2.1.1 for AIX TCP/IP Client Example	183
40. CAE 2.1.1 for OS/2 or SDK 2.1.1 for OS/2 TCP/IP Client Example	185
41. APPC Client Configuration	192
42. APPC Client Planning Template	193
43. TPN Profile Screen	195
44. TPN Profile for AIX Client	196

45.	TPN Profile Screen	208
46.	TPN Profile for AIX Client	210
47.	Partner LU Profile, Conversation Security Level	211
48.	DB2 for MVS DRDA AR Configuration	215
49.	DB2 for MVS DRDA AR Configuration Template.	216
50.	DataJoiner Instances.	227
51.	IBM's Data Replication Solution and Product Environment	235
52.	DataJoiner with the replication administration tool Scenario	238
53.	The Replication Administration Tool's Primary Window.	240
54.	DataJoiner in Action	243
55.	Capture Triggers at the Source Server	245
56.	The Apply Program	248
57.	Sample apply_name.ini File	252
58.	Sample Server Options for Sybase	252
59.	Sample Server Options for Microsoft SQL Server	252
60.	The CCD Table	256
61.	Transaction-Based versus Transaction-Consistent Replication	257
62.	The Apply Program Push Scenario	262
63.	The Apply Program Pull Scenario	263
64.	DataJoiner and the Replication Administration Tool on a Single Windows NT Workstation	271
65.	DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses DataJoiner to Connect.	271
66.	DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses the Replication Administration Tool to Connect.	272
67.	DataJoiner Maps the Replication Administration Tool User IDs to Source and Target.	278
68.	DataJoiner Uses the Apply Password File to Connect to Source and Target	280
69.	The Replication Administration Tool Primary Window	288
70.	The Preferences Notebook: Connection	290
71.	Creating Replication Control Tables	291
72.	The Replication Administration Tool Primary Window	295

Tables

1. Memory and Disk Requirements for DataJoiner	22
2. DataJoiner Data Sources	27
3. Common Migration Inhibitors and Recommended Solutions	40
4. Data Sources and Their Access Methods	69
5. Data Sources Supporting Two-Phase Commit	169
6. The Apply Program Cycle: A High-Level Overview of How a Subscription Set is Processed.	251
7. Promote Functions	265
8. Supported Sources and Targets	269
9. Summary of Capture and Apply Programs Supported by DataJoiner with the Replication Administration Tool	273
10. Configuring Database Connections in DataJoiner: AIX.	284
11. Data Type Conversion: DB2 to Oracle	300
12. Data Type Conversion: DB2 to Informix	300
13. Data Type Conversion: DB2 to MS SQL Server, Sybase, or SQL Anywhere	301
14. Data Type Conversion: DB2 to Microsoft Jet	302
15. DataJoiner DRDA Application Server Function	307
16. Bind Options Supported by the DataJoiner DRDA Application Server	308
17. CCSID Support for DataJoiner for AIX DRDA Application Server	312
18. OS/400 PTFs required for DRDA AS	318
19. Supported Code Page Conversions	336
20. Supported Locales, Code Sets, Code Pages, and CCSIDs	338
21. Relationships Between SBCS and DBCS Code Pages	341
22. Relationships Between EUC Code Pages and Character Groups	342
23. Supported Code Sets	343
24. Code Page Mappings	344
25. Mixed Character Set Code Points	345
26. Japanese EUC Code Points	345
27. Traditional Chinese EUC Code Points	346
28. Korean EUC Code Points	346
29. Simplified Chinese EUC Code Points	346
30. Formats for String Representations of Dates	350
31. Formats for String Representations of Times	350
32. Date and Time Formats by Country Code	352
33. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner	361

Welcome to DataJoiner

This book provides you with planning information and installation and configuration procedures for DataJoiner for AIX. 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 provided by DataJoiner and lists the new features for this version.

Part 2 provides information about planning for and installing DataJoiner.

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

Part 4 provides information about configuring communications between DataJoiner and its clients.

Part 5 describes how to maintain DataJoiner.

Part 6 contains information about replication, including planning, configuring, defining, replication sources and targets, and maintaining replication objects.

Part 7 contains information about configuring DataJoiner as a DRDA application server (AS), national language support, and available education and service for DataJoiner.

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.

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.

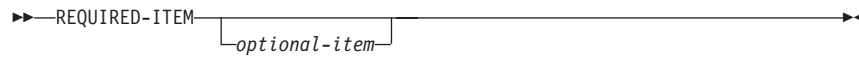
Required items

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

▶▶—REQUIRED-ITEM—▶▶

Optional items

Optional items appear below the main path.

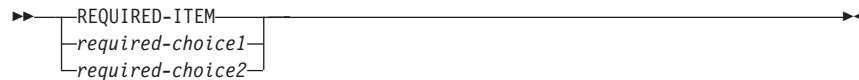


If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.



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.

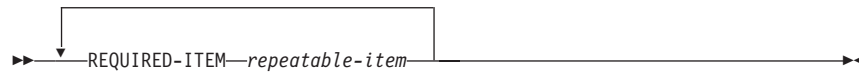


If choosing one of the items is optional, the entire stack appears below the main path.

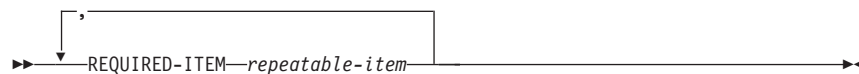


Repeatable items

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.



Road Map



Use the following table to find the information you need quickly.

If you want to...	Refer to...
Understand the relationship between DataJoiner and related products,	"Chapter 1. About DataJoiner" on page 3.
Find out about new features and functionality included in this release,	"Chapter 2. What's New in DataJoiner Version 2?" on page 15.
Confirm that your system meets DataJoiner hardware and software prerequisites,	"Hardware Requirements" on page 21 and "Software Requirements" on page 25.
Install DataJoiner using smitty or installp,	"Installation" on page 44.
Configure DataJoiner to communicate with your IBM, Oracle, Sybase, Informix, Microsoft, and other supported data sources,	"Chapter 6. Configuring Access to Data Sources" on page 69.
Install the client products for Macintosh, AIX, HP-UX, Solaris, SINIX, and DOS,	The <i>Installing and Using Clients</i> book for each of these platforms. These books are included on the Client Pack CD-ROMs in softcopy format.
Use the DataJoiner components,	"Chapter 5. Setting up an Instance" on page 55 and "Chapter 22. Maintaining the DataJoiner Environment" on page 219.
Plan, configure, administer, and operate DataJoiner's replication administration tool,	"Part 6. Replicating Data" on page 231.
Set up and use the DRDA Application Server feature to have host applications access LAN data,	"Appendix A. DRDA Application Server" on page 307 .
Learn about last-minute changes to DataJoiner,	The README files that are on the CD-ROMs or that are installed with the products.
Configure the DB2 Spatial Extender,	The <i>IBM DB2 Spatial Extender Administration Guide and Reference</i> included in hardcopy and on the DataJoiner CD-ROM.

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 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 (Software Developer's Kit) 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:

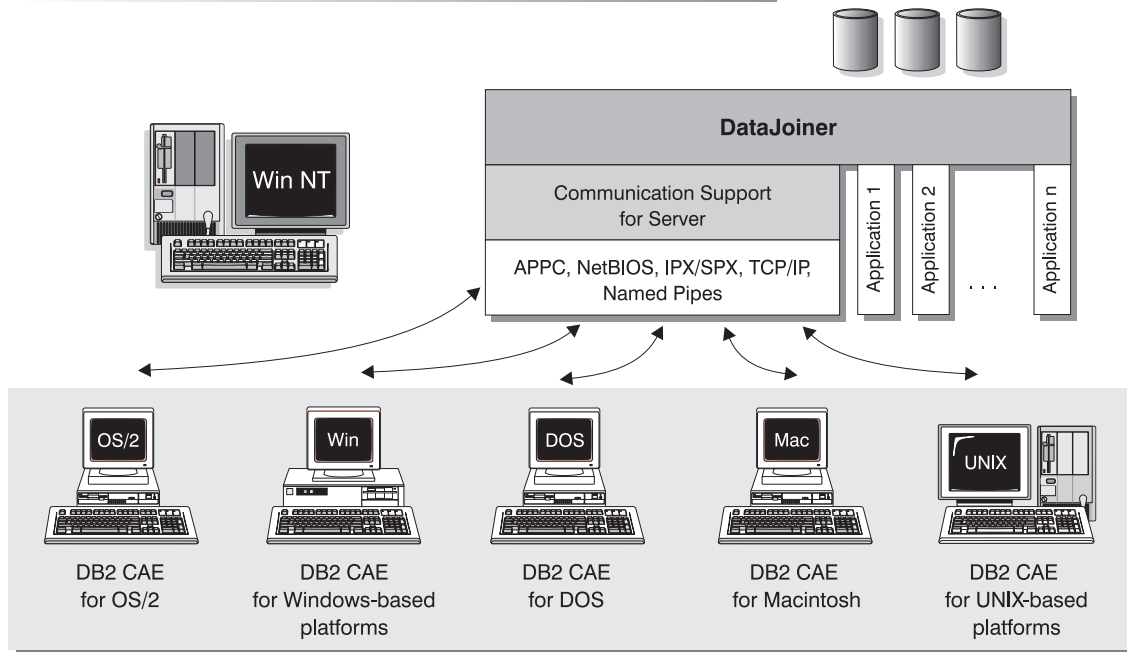
- OS/2
- Windows NT, Windows 95, Windows 3.x
- UNIX (AIX, HP-UX, Solaris, SINIX)
- Macintosh
- DOS

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.

DataJoiner Remote Client Support



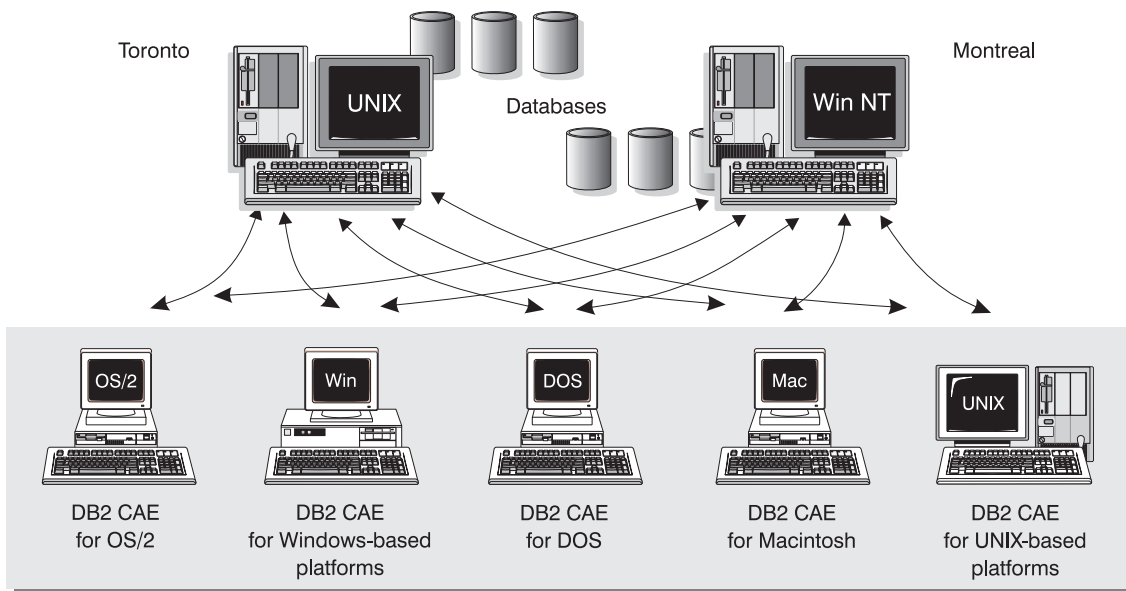
NOTE: CAE is Client Application Enabler. Not all protocols are supported for all the clients.

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).

The information in "Part 4. Connecting Clients to DataJoiner" on page 173 shows you how to use Database Director to manage your databases and directories. The online help for Database Director contains additional 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.

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 AIX. 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 Internet at URL <http://www.software.ibm.com>. (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

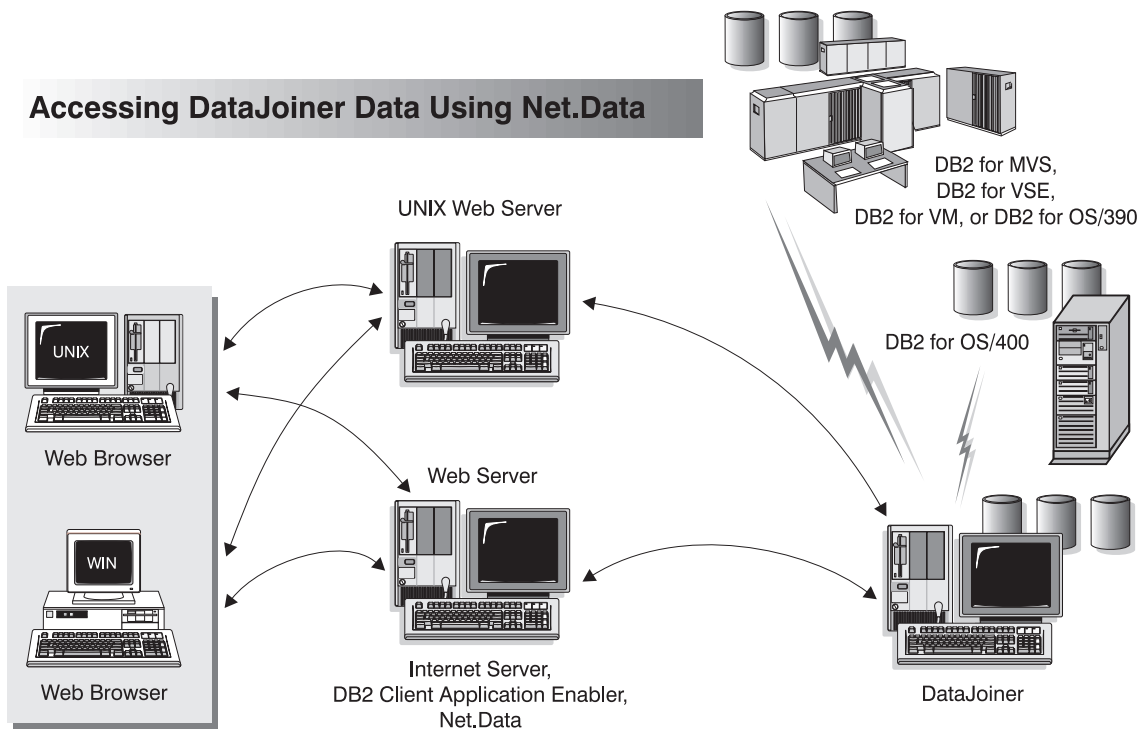


Figure 3. Net.Data

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.

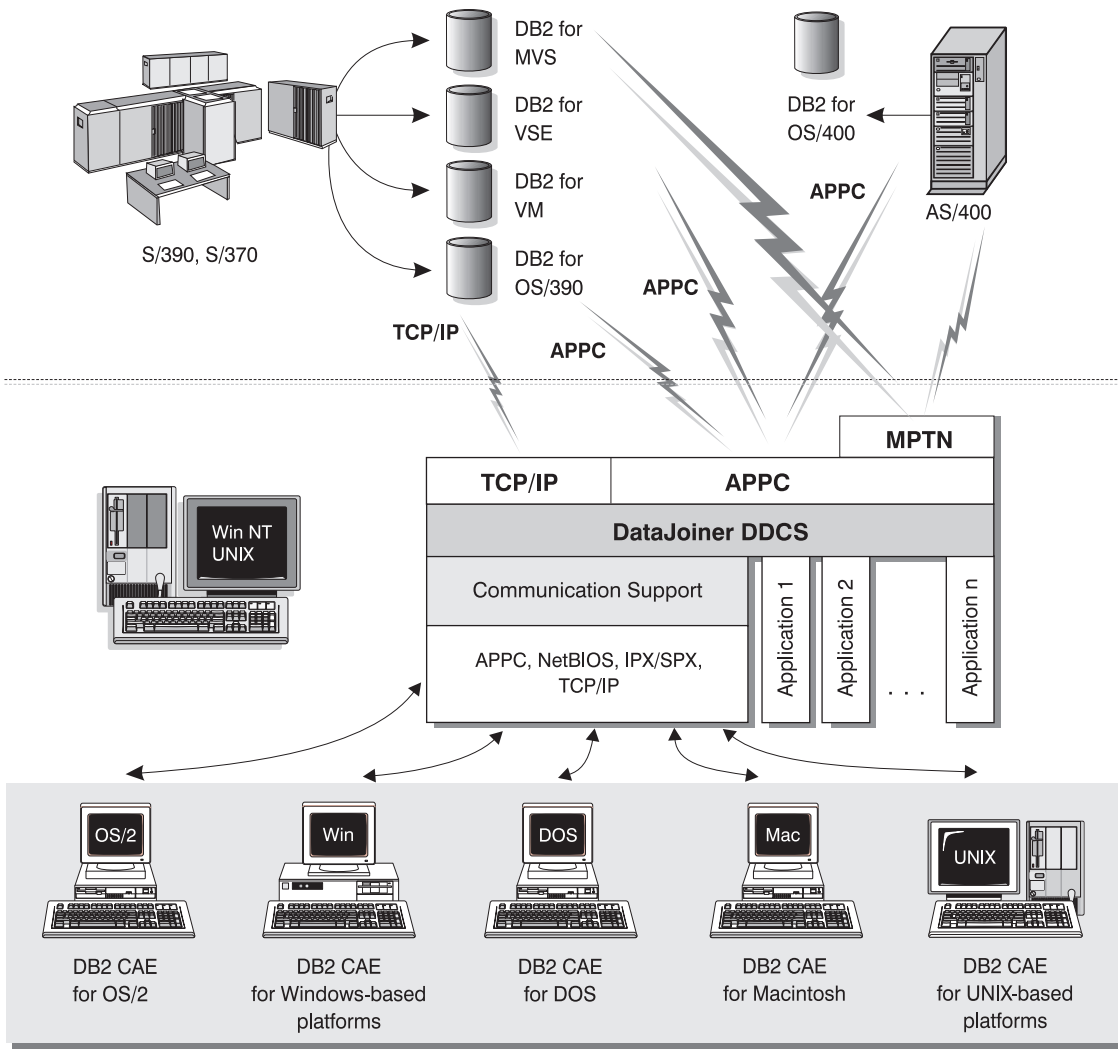
DataJoiner DDCS provides applications with transparent online access to data by implementing a standard architecture for managing distributed data. This standard is known as Distributed Relational Database Architecture (DRDA). For example, you can use DataJoiner DDCS with:

- 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.

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 6. Configuring Access to Data Sources" on page 69 for more information.

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

DataJoiner Version 2 offers new features and enhancements. They include:

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

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

DataJoiner for Windows NT

DataJoiner has extended its reach to provide industrial strength heterogeneous database management on Windows NT systems. DataJoiner for Windows NT supports the same SQL and features as DataJoiner for UNIX-based platforms.

Support for Oracle 8, RDB, and SQL Anywhere

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

- Oracle 8 (on any system that DataJoiner accesses from AIX or Windows NT)
- Oracle RDB Version 6 and above (on any system that DataJoiner accesses from Windows NT)
- Sybase SQL Anywhere Version 5.0 (on any system that DataJoiner accesses from Windows NT)

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 now allows 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.

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

DataJoiner has many options and components that you might want to use in your environment. This chapter describes these options and components. It also lists the requirements that are needed to install DataJoiner and to use its various options and components.

Hardware Requirements

This chapter identifies the hardware requirements for DataJoiner only. It does not include hardware requirements for products that DataJoiner accesses, such as Informix**, Oracle**, or Sybase**, nor does it contain hardware requirements for remote systems where clients reside. Requirements for DataJoiner's replication components, the replication administration tool and Apply, are provided in "Part 6. Replicating Data" on page 231. Classic Connect requirements are provided in the *DataJoiner Classic Connect Planning, Installation, and Configuration Guide*.

General Requirement Information

The minimum hardware configuration for DataJoiner is a machine that is capable of:

- Reading a CD-ROM.
- Supporting the appropriate version of AIX, as documented in "Software Requirements" on page 25.

When you estimate your hardware requirements for DataJoiner, you need to consider both:

- The fixed-disk requirements for installing product files on the target machine
- The fixed-disk requirements and memory requirements for each instance of the product

An *instance* is a DataJoiner server, and you can have more than one instance on the same machine. Additional resource is required for each instance of DataJoiner that is created, each database, and each client. You can read more about instances in "Chapter 5. Setting up an Instance" on page 55.

Memory and Disk Requirements

This section summarizes the memory requirements and disk requirements that are related to DataJoiner.

To estimate the memory requirements and disk requirements for a particular configuration, add the recommended memory size for each installed function and application that you plan to run concurrently.

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.

Using less 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.

Recommended Memory Sizes

Table 1 provides the recommended memory size in Megabytes (MB) for concurrent operation of various functions and the recommended disk size in MB. Use the table to calculate your requirements.

Table 1. Memory and Disk Requirements for DataJoiner

Functions	Recommended Working Set Memory (MB)	Recommended Disk Space (MB)
Requirements for Product Files^a		
DataJoiner base (dix)	N/A	60.0
DRDA AR AS (drda_ar_as)	N/A	1.1
Classic SNA (dix_classic_sna)	N/A	.22
SNA clients (dix_sna_clients)	N/A	.17
DBA Tool (dix_dba)	N/A	1.1
DBA Tool messages per locale (msg.*.dba)	N/A	.9
Messages per locale (msg.*.dix)	N/A	1.1
Requirements for each Instance of DataJoiner		
Instance Base ^b	buffer ^c + 2.7	.75 ^d
Instance Base for Classic Connect ^b	buffer ^c + 5.0	.75 ^d
For each additional concurrent database	buffer ^c + 1.7	1.1 ^d
<i>DataJoiner Instance Requirements for each Client Connection</i>		
For each concurrent local user or application	1.0	N/A
For each concurrent spatial application	n ^e * 1.2	N/A
For each concurrent remote client	.75	N/A
For each Informix data source accessed	.85	N/A
For each Classic Connect data source accessed	2.5	N/A
For each other data source accessed, other than Informix or Classic Connect	.18	N/A
TOTAL (for your requirements)		

Notes:

- a 15 MB of additional temporary space is required during installation.
- b Assumes one process using one local database on one instance.
- c Additional memory for some database configuration parameters, including the database buffer pool or the sort heap, might be required depending on your workload. The buffer pool default is 4MB. If you decide later that the buffer size needs to change, see the *DB2 Administration Guide* for instructions on how to change it. The buffer pool applies to locally stored DataJoiner data only, not data that is retrieved from data sources. Keep this fact in mind when you read instructions in the *DB2 Administration Guide*.

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 were accepted.
- d The extra disk space for each database is for table definitions and internal structures for each database and does not include user data. The extra disk space is subject to many variables, so actual requirements might differ.

The default configuration allocates three log files, each 4MB in size. These logs can be relocated to another file system and can be made larger or smaller as required. See the *DB2 Administration Guide* for more information.
- e n is the maximum number of ADT parameters that are passed to a spatial function in your query. For example, the query `SELECT cust_id FROM customers WHERE within (loc, buffer(loc, 100))` has four parameters (100, loc, buffer(loc,100), and loc). If this is the largest query in your application, the amount of working set memory required is $4 * 1.2 = 4.8\text{MB}$.

This memory requirement is specified by the environment variable **UDF_MEM_SZ**. The recommended setting for this variable is 2048 (8MB).

Calculating Memory Requirements

If there is not enough real memory on a DataJoiner server, DataJoiner runs in virtual memory that is provided by the operating system. In this case, performance drops significantly because virtual memory requires disk I/O time to page in and out of real memory. Be sure to consider the memory requirements for your operating system and other applications that will run concurrently with DataJoiner when estimating the requirements for your machine configuration.

You can calculate the real memory requirements for DataJoiner with your specific configuration by adding together the memory requirements for all instances that will run concurrently. Use one of the following formulas to calculate the memory requirement (in MB) for a specific instance.

General form:

Instance base +
Additional concurrent databases +
Additional concurrent users +
Data sources

Formula in more detail:

$$\text{bmem} + \text{buffer} + (\text{dnum} - 1) * (1.7 + \text{buffer}) + (\text{cnum} - 1) * \text{cmem} + \text{cnum} * (\text{smem}[1] + \text{smem}[2] + \dots + \text{smem}[s])$$

where:

bmem Is the memory for instance base in Table 1 on page 22. The requirement of instance base for Classic Connect is used in Table 1 on page 22.

buffer Is the buffer space referred to in footnote c of Table 1 on page 22.

dnum Is the number of concurrent databases. Note that the instance base requirement already contains the requirement for one database.

1.7 Is the memory requirement for an additional concurrent database in Table 1 on page 22 .

cnum Is the number of concurrent local users, local applications, or remote clients.

cmem Is the memory requirement for one local user, local application, or remote client.

smem[1] ... smem[s]

Is the memory requirement for the data sources accessed. Summarize each data source one by one. Note in Table 1 on page 22 that the requirement for Informix and Classic Connect data sources is different from that of other data sources.

For example, assume that there are 20 concurrent applications running on a DataJoiner server. These 20 applications use one DataJoiner instance and two databases. The instance is configured to access Classic Connect. The applications access one Oracle, one Informix, and one Classic Connect data source. The database buffer size referred to in footnote c is 4 MB.

Memory requirements for this example are:

$$\text{bmem} + \text{buffer} + \text{dnum} * (1.7 + \text{buffer}) + (\text{cnum} - 1) * \text{cmem} + \text{cnum} * (\text{smem}[1] + \text{smem}[2] + \dots + \text{smem}[s])$$

or

$$6.7 + 4.0 + (2 - 1) * (1.7 + 4.0) + (20 - 1) * 1.0 + 20 * (0.18 + 0.85 + 2.5)$$

which equals 106MB.

Software Requirements

This chapter identifies the software requirements for DataJoiner only. It does not include requirements for prerequisite software or data sources that DataJoiner accesses, such as Informix, Oracle, or Sybase, nor does it contain requirements for remote systems where clients reside.

Software requirements for the replication administration tool and Apply are provided in *IBM DB2 Replication Guide and Reference: V5*.

Operating System Requirements

DataJoiner requires one of the following operating systems:

- IBM AIX Version 3.2.5 with the following PTFs or any PTFs that supersede any of the following PTFs:
 - U403173
 - U412397
 - U412815
- IBM AIX Version 4.1 or higher

Spatial Extender Requirement:

To take advantage of the DB2 Spatial Extender option (djx_02_01_01.se), AIX Version 4.1.4 or higher is required.

If you have installed the IBM RISC System/6000 feature for common mode development (FCM) static libraries, and you are experiencing problems with DataJoiner, please contact IBM Service.

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. 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 through any of the supported communication protocols that are documented

in “Part 4. Connecting Clients to DataJoiner” on page 173. 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.

DRDA Databases

In addition to other DRDA-compliant database management systems, DDCS for AIX and DataJoiner for AIX support the following IBM host relational database products:

- **DB2 for MVS/ESA Version 2 Release 3**

- **PTF UN75958**

- Required to use DB2 for MVS/ESA as an application requester.

- **PTF UN54600**

- Required for connection with AIX.

- **PTF UN56735**

- Required for connection with AIX.

- **DB2 for MVS/ESA Version 3 Release 1**

- **PTF UN75959**

- Required to use DB2 for MVS/ESA as an application requester.

- **PTF UN54601**

- Required for connection with AIX.

- **PTF UN56736**

- Required for connection with AIX and two-phase commit with CICS/6000.

- **PTF UN73393**

- Required to use two-phase commit with CICS/6000 and to connect to DB2 for MVS/ESA using DCE Services from AIX clients.

- **DB2 for MVS/ESA Version 4 Release 1**

- **APAR PN69689**

- Required to use DRDA stored procedures.

- **APAR PN69748**

- Required to use DRDA stored procedures.

- **DB2 for OS/390**

- **SQL/DS Version 3 Release 3 (DB2 for VSE and VM)¹**

- **PTF PN43497**

- Required for connection with AIX.

- **PTF UN47865**

- Required to use DB2 for VSE and VM as an application requester.

- **APAR PN69073**

- Required to use DB2 for VSE and VM as an application requester.

Additional scripts must be run on the OS/2 or AIX application server. See the *DB2 Installation and Operation Guide* for more information.

- **SQL/DS Version 3 Release 4 (DB2 for VSE and VM)¹**
- **DB2 for OS/400 Version 2 Release 2¹**

PTF SF13747

Required for connection with AIX.

PTF SF13748

Required for connection with AIX.

DB2 for OS/400 Version 3 Release 1¹

Note: You should consult with your host database coordinator to determine that the appropriate prerequisites and security considerations have been considered and installed to support the AIX environment and host database environments.

Requirements for DataJoiner Data Sources

DataJoiner for AIX can access the data sources in Table 2 and their subsequent releases.

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 • IBM DB2 for OS/400 V4R2 	included in DataJoiner

¹. The replication administration tool currently does not support this data source.

Table 2. DataJoiner Data Sources (continued)

Data Source	Client Software Needed on the DataJoiner System
IBM SQL/DS V3R3 with APAR PN43497 or higher	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 higher; 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
IMS V2	DataJoiner Classic Connect
Informix V5 or V6	Informix Net V5.03 or higher and ESQL/C V5.03 or higher libraries.
Informix OnLine V7.1 (non-GLS, -XPS, and -SE versions only)	Informix ESQL/RT or equivalent libraries
Microsoft SQL Server V4.21	One of the following: <ul style="list-style-type: none"> Sybase Open Client Visigenic 1.1 or higher (for AIX 3.2.5) Visigenic 2.0 or higher (for AIX 4.1 and above)
ODBC, X/Open CLI-compliant data sources through the generic access API	ODBC driver
Oracle V7.0.13	Oracle SQL*Net V1 or V2 or Oracle Net8
Oracle V8	Oracle SQL*Net V1 or V2 or Oracle Net8
Sybase V4R6	Sybase Open Client
VSAM	DataJoiner Classic Connect
Other non-relational data sources	Cross Access Corporation's CrossAccess Data Delivery System

Communication Protocol Requirements

DataJoiner supports the APPC, IPX/SPX, and TCP/IP network protocols for server function; APPC and TCP/IP are supported for client function. The communication software you need depends on the protocol you wish to use. The database server can accept requests from multiple protocols concurrently. If you wish to use several protocols, ensure that you have all the required products installed.

If you have IBM AIX Version 3.2.5 installed, you will need one of the following protocols:

- For APPC connectivity, IBM SNA Server/6000 Version 2.1 or later

Note: If you plan to use DRDA Application Server, you will need to install IBM SNA Server/6000 Version 2.1.1 with PTFs U435033 and U435034.

- For IPX/SPX connectivity, IBM AIX NetBIOS and IPX Support/6000 Version 1.1
- For TCP/IP connectivity, the TCP/IP component of the operating system must be installed

If you have IBM AIX Version 4.1 installed, you will need one of the following protocols:

- For APPC connectivity, either of the following products:
 - IBM SNA Server/6000 Version 3.1 or later
 - IBM Communications Server Version 4 or later
- For IPX/SPX connectivity, one of the following products:
 - IBM AIX NetBIOS and IPX Support/6000 Version 2.1, or
 - IBM NetWare for AIX Version 3.11B. (This is the complete product which includes a NetWare file server)
- For TCP/IP connectivity, the TCP/IP component of the operating system must be installed.

Graphical User Interface Requirements

To use the Database Director tool, you must have the following software installed on your system:

- IBM AIXWindows Version 1.2.3 or later
- Motif Version 1.2 (x11r5)

Transaction Managers

DataJoiner functions as a transaction manager, or you can install any of the following supported transaction managers:

- IBM CICS/6000 Version 1.2 or 2.1
- Encina TP Monitor Version 1.3
- Tuxedo System/T Version 4.2.2

To support DRDA two-phase commit with Encina TP Monitor or CICS/6000 against DB2 for MVS/ESA Version 3.1 or DB2 for OS/400 Version 3.1, you must also install Encina PPC Gateway Version 1.3 with APARs IX49555 and IX50207.

Two-phase commit restriction:

Tuxedo System/T cannot be used for two-phase commit with DRDA databases.

High Availability using HACMP

The IBM AIX High Availability Cluster Multi-Processing/6000 (HACMP/6000) system enables clustered RISC System servers to recover from server, disk, network, or network interface failures. To use HACMP with DataJoiner, install IBM HACMP/6000 Version 2.1 or later on your system.

DB2 SNMP Subagent

Simple Network Management Protocol (SNMP) products include IBM NetView for AIX and IBM NetView for OS/2. DataJoiner provides support for SNMP management products through the DataJoiner SNMP Subagent, included as a component of the DataJoiner server. SNMP management products, such as NetView, allow centralized management of a system's hardware and software components. The DataJoiner SNMP Subagent allows DataJoiner servers to be managed just like any other SNMP-managed resource.

See the readme file for the software requirements for the SNMP Subagent.

Two-Phase Commit Requirements

DataJoiner supports two-phase commit for Oracle, Informix, Sybase (ctlib only), and various versions of DB2. The XA libraries must be installed and activated for any data source participating in a two-phase commit transaction. Additionally, the XA libraries must be link-edited with the DataJoiner server. These requirements do not apply to DB2-family data sources.

See "Chapter 15. Configuring for Two-Phase Commit" on page 169 for more information.

Supported Compilers for DataJoiner

DataJoiner provides support for C, C++, COBOL, and FORTRAN 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 AIX provides C, C++, and FORTRAN 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 AIX include the following:

- IBM XL C Version 1.2.1 or 1.3
- IBM C for AIX Version 3.1
- IBM C Set++ Version 2.1
- IBM C Set++ for AIX Version 3.1

- IBM AIX XL FORTRAN Version 2.3, 3.1, or 3.2
- IBM COBOL Set for AIX Version 1.1
- Micro Focus COBOL Version 3.1.49 or later
- IBM REXX/6000 (available as PRPQ 5764-057)

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.

ADSTAR Distributed Storage Manager

If you plan to use the ADSTAR Distributed Storage Manager facilities for backup and restore of your databases, you must have the ADSTAR Distributed Storage Manager product installed on your server. The ADSTAR Distributed Storage Manager API Version 1.2.7, which allows the database manager to interface with the host product, is included with DataJoiner.

Information and support for ADSM and its APIs are provided by ADSTAR. To contact ADSTAR from the U.S. or Canada, call 1-800-4-ADSTAR (1-800-423-7827). Outside of the U.S. and Canada, use the World Trade access number, which is 1-408-284-0385.

Prerequisites for DataJoiner Options

Many of the options that are contained on the DataJoiner for AIX Version 2 CD-ROM require you to select prerequisite options or install prerequisite software. This section lists options on the DataJoiner for AIX Version 2 CD-ROM and their prerequisite requirements. The options are similar to DB2 Version 2 options.

If you are using smitty to install DataJoiner, you will need to scroll right to see an option's abbreviated name (djk_02_01_01.xxx).

DB2 Client Application Enabler (djk_02_01_01.client)

- Prerequisite
 - IBM AIX (R) Version 3.2.5
 - or
 - IBM AIX (R) Version 4.1

On AIX 4.1 systems, the following apply:

 - xIC.rte 3.1 or higher must be installed.
 - On AIX 4.1.4 or higher systems, dce.pthreads.rte 2.1.0 or higher must be installed.
 - DCE directories are supported only on AIX 4.1.4 or higher.
- Selectable options
 - None

The message catalog for US English is included.

Command Line Processor Option (djx_02_01_01.clp)

- Prerequisite
 - djx_02_01_01.client
- Selectable options
 - None

SDK Options (djx_02_01_01.sdk...)

- Prerequisite
 - djx_02_01_01.clp
- Selectable options
 - djx_02_01_01.sdk.c (C include files and sample programs)
 - djx_02_01_01.sdk.fortran (Fortran include files and sample programs)
 - djx_02_01_01.sdk.cobol (COBOL include files and sample programs)
 - djx_02_01_01.sdk.cli (Call Level Interface samples)
 - djx_02_01_01.sdk.misc (SDK utilities and samples)

DataJoiner Executables Options (djx_02_01_01.db2)

- Prerequisite
 - djx_02_01_01.clp
- Selectable options
 - djx_02_01_01.db2.rte (DataJoiner executables)
 - djx_02_01_01.db2.misc (DataJoiner utilities and samples)

Code Page Conversion Option (djx_02_01_01.conv)

- Prerequisite
 - djx_02_01_01.client
- Selectable options
 - None

Communication Support Options (djx_02_01_01.cs)

- Prerequisite
 - djx_02_01_01.db2.rte
- Selectable options
 - djx_02_01_01.cs.rte (Base with TCP/IP support)
 - Prerequisite
 - djx_02_01_01.db2.rte
 - djx_02_01_01.cs.sna (SNA support)
 - Prerequisites

- djx_02_01_01.cs.rte
- SNA Server/6000 Version 2.1
- or
- Desktop SNA Version 1 (AIX 4.1 only)
- djx_02_01_01.cs.drda (DRDA Application Server)
- Prerequisites
 - djx_02_01_01.cs.sna
 - SNA Server/6000 Version 2.1.1 with PTFs U435033 and U435034
- djx_02_01_01.cs.ipx (IPX/SPX support)
- Prerequisites
 - djx_02_01_01.cs.rte
 - IBM Netware for AIX Version 3.11B
- or
- AIX NetBIOS and IPX Support/6000 Version 1.1

DDCS Options (djx_02_01_01.ddcs)

- Prerequisite
 - SNA Server/6000 Version 2.1
 - or
 - Desktop SNA Version 1 (AIX 4.1 only)
 - or
 - Encina Server for AIX Version 1.3 (encServ.obj 1.3 on AIX 3.2 or AIX 4.1 equivalent)
- Selectable options
 - None

DB2 Database Director Option (djx_02_01_01.dd)

- Prerequisites
 - djx_02_01_01.clp
 - IBM Information Presentation Facility
 - X11fte.motif1.2.obj (1.2.3 or higher on AIX 3.2 only)
- Selectable options
 - None

DB2 Visual Explain Option (djx_02_01_01.ve)

- Prerequisite
 - djx_02_01_01.dd
- Selectable options
 - None

| **DB2 Spatial Extender Option (dix_02_01_01.db2se)**

- | • Prerequisite
 - | – IBM AIX (R) Version 4.1
- | • Selectable options
 - | – None

Chapter 4. Installing DataJoiner

This chapter covers installation planning, a common AIX installation problem, and installation.

Installation Planning

This section covers prerequisite planning and other considerations.

Before installing DataJoiner, be sure that you understand DataJoiner's hardware and software requirements. See "Hardware Requirements" on page 21 and "Software Requirements" on page 25.

This section discusses the tasks you need to perform and the decisions you need to make before installing DataJoiner.

Related Reading:

The README.CD file on the DataJoiner CD-ROM contains the most up-to-date installation notes. Read this file before attempting to install DataJoiner.

Install Prerequisite Software

After familiarizing yourself with DataJoiner's software requirements, make sure that any prerequisite software is installed. For example, you might need client software for access to data sources like Sybase and Oracle.

Understand DataJoiner Security

Security is a complex issue because DataJoiner accesses so many data sources. Read the entire security section of the *DataJoiner Administration Supplement* before installing the product. Doing so will familiarize you with issues that might affect the installation choices you make.

Decide on Appropriate Authorization

Be sure to have a plan in mind for setting up proper access to your databases. A key part of this plan is to authorize users according to their responsibilities. Before installing and configuring DataJoiner, consider what authorization levels you will grant to certain users. For more information, see the *DataJoiner Administration Guide*.

Decide Whether You Need Additional Password Encryption

Decide whether passwords that are stored in the SYSCAT.REMOTEUSERS catalog view need additional encryption.

DataJoiner might need passwords to access data sources. If so, you can store the passwords in the system catalog.

DataJoiner encrypts passwords when they are added to the catalog and decrypts them just before attempting to access data sources. Users and applications are never allowed to see passwords.

Every organization must decide what level of encryption meets its needs. If yours does not need additional encryption when DataJoiner is installed, you can add it later. If you add it during installation but decide later to remove it, you can do that too. However, changing encryption poses significant implications for existing encrypted passwords in the system catalog.

Do not confuse passwords stored in SYSCAT.REMOTEUSERS with those stored in the Apply passwords file. Apply passwords cannot be encrypted.

Decide Which Client Protocols to Use

You need to determine which communication protocols your clients will use to communicate with DataJoiner. The following table summarizes protocols that are supported by DataJoiner for communication with clients and points you to detailed configuration instructions:

Protocol	For more information, see...
TCP/IP	"Chapter 17. TCP/IP Support for Remote Clients" on page 177.
IPX/SPX	"Configuring IPX/SPX for DB2RA Data Sources" on page 104.
APPC	"Chapter 19. APPC Support for Remote Clients" on page 191.

Support for each of these protocols is enabled during installation by either installing the entire DataJoiner product, which would enable all three protocols, or by selecting only specific protocol options.

Replication Considerations

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

- The replication administration tool runs on a Windows 95 or Windows NT workstation and accesses DataJoiner through the DB2 CAE for Windows 95 or NT. This means that you will need to configure the DataJoiner instance for remote client access.
- 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 6. Configuring Access to Data Sources" on page 69 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 8. Accessing DB2RA Data Sources” on page 101 for more information.

- To access IBM IMS/ESA non-relational databases, you need DPROP NR.
- To access DB2 for OS/390 data sources, the Apply program should go through DataJoiner DDCS, while the replication administration tool can use either DataJoiner DDCS or DDCS 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 for OS/390 than going to another workstation to access DataJoiner DDCS and subsequently DB2 for OS/390. 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 file space and memory impact. When replicating to one or more non-IBM relational databases, all targets can be accessed through a single DataJoiner database, thus saving resources.
- When replicating from non-IBM databases to DB2 for OS/390 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 as described in “Chapter 21. Configuring the DataJoiner DRDA Application Server and Clients” on page 207.
- 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).² 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 DataPropagator Relational V5 Capture program.

Replication planning and configuration information are discussed more fully in “Part 6. Replicating Data” on page 231.

Migrating From Previous Releases

This section shows you how to migrate a previous release of DataJoiner to Version 2.1.1.

CAUTION!

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

Do not uninstall an existing version of DataJoiner until your migration work is completed and tested.

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

The migration process consists of two distinct, but interrelated, steps:

1. Instance Migration
2. Database Migration

Instance Migration

To prepare for instance migration:

1. Login as **sysadm** for the instance you are migrating.
2. Ensure that all local databases that you wish to migrate are cataloged before running the **db2imigr** command.
3. End all applications that are currently using the instance.
4. Stop the database manager and the command line processor.
5. Issue the **db2stop** and **db2 terminate** commands.

Login as root to perform the remaining steps.

Create a group and user for Fenced User Defined Functions (optional). This is described in “Step 1: Create an Instance of the Product” on page 55.

The **db2imigr** script is provided to migrate an existing instance to Version 2.1.1. The instance migration routine is invoked from the `/usr/lpp/djx_02_01_01/instance` directory as follows:

```
▶▶ db2imigr -instanceName -a server -uUdfName
```

```
graph LR; A["-a server  
-a client  
-a dcs"]; B["-uUdfName"]; C["▶▶ db2imigr -instanceName"]; C --- A; A --- B; B --- D["▶▶"]; style A fill:none,stroke:none; style B fill:none,stroke:none; style C fill:none,stroke:none; style D fill:none,stroke:none;
```

where:

instanceName

Represents the name of the existing instance to be migrated.

- a** Indicates the authentication type for the new instance. Valid authentication types are SERVER, CLIENT, and DCS. If the **-a** parameter is not specified, then the authentication type will default to SERVER.

Note: The authentication type of the instance applies to all databases under the instance.

- u** Indicates the AIX user and group under which Fenced UDFs will execute.

Restriction:

UdfName may not be **root** or **bin**.

If *UdfName* is unspecified, the user and group default to nobody:nobody; this means that there are no file access permissions defined for UDFs or DARIs.

You may set the *UdfName* later with the **db2iupdt** command, if you do not do it now.

While **db2imigr** does not automatically initiate the migration of instances, it checks to detect conditions that would prevent the successful migration of any of the local databases that are cataloged in the instance. If it detects any of these conditions, then **db2imigr** aborts the instance migration and generates a report that lists the conditions that were detected. The report is stored in a file called `INSTHOME/migration.log`.

An error is logged for each database that 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.

A warning is also logged for each local database entry with an authentication type that is different from the authentication type specified on the **db2imigr** command. While this will not stop the migration, you may still want to take some action. See "Correcting Migration Inhibitors" on page 40 for suggested corrective actions, correct all the errors that are reported, and run **db2imigr** again.

When the **db2imigr** verification finds no errors, the instance migration will be initiated.

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 in `/usr/lpp/djx_02_01_01/bin` as follows:

```
▶▶—db2ckmig—database—/l filename—————▶▶
```

database

Specifies an alias name of a database to be scanned

/I 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" for suggested corrective actions).

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

Correcting Migration Inhibitors

The following table shows common inhibitors to successful database migration and prescribes the 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 .
Database inconsistent	Restart the database to return it to a consistent state.

See the (Version 1) *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). **db2imigr** does not perform any verification of uncataloged databases.

Migration Processing

The instance migration process performs the following actions:

- Backs up the down-level instance.
A Version 1.2 instance is moved from `INSTHOME/sqllib` to `INSTHOME/sqllib_v1`, and a Version 2.1 instance is moved from `INSTHOME/sqllib` to `INSTHOME/sqllib_v2`. Once you are certain that you no longer require the down-level instance you can delete the `INSTHOME/sqllib_v1` or `_v2` directory.
- Creates a Version 2.1.1 instance.
Relevant information from down-level DataJoiner directories and configuration files is copied and merged with appropriate Version 2.1.1 defaults.
- Updates the list of instances.
- Copies the down-level version's `db2profile` and `db2cshrc` files to `db2profile.v2` and `db2cshrc.v2` under the Version 2.1.1 instance.
- Copies the contents of the down-level version's function directory, `INSTHOME/sqllib/function`, to the Version 2.1.1 instance.

Reversing Instance Migration

If for some unexpected reason you must reverse the migration, login as root and issue the **db2imigrev** command as follows:

```
db2imigrev instanceName
```

Note: Databases that have been migrated to Version 2.1.1 are unusable to Version 1.2 or 2.1.

This will restore the `INSTHOME/sqllib_v1` or `_v2` directory to `INSTHOME/sqllib` and create a copy of the Version 2.1.1 `INSTHOME/sqllib` to `INSTHOME/sqllib_v211`.

You can now work with Version 1.2 or 2.1 databases that remain. To return to Version 2.1.1, execute **db2imigr** again; this restores Version 2.1.1 of DB2 from the copy (`INSTHOME/sqllib_v211`).

Database Migration

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.

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 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 SQLEMGDB migrate database API
 - 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.

Other Considerations

The information in this section is helpful when planning the installation of DataJoiner.

Developing Applications with Micro Focus COBOL

If you plan to develop DataJoiner applications with the Micro Focus** COBOL precompiler front end, runtime interpreter, or debugger (the "Animator"), you must add the DataJoiner Generic API entry points to the Micro Focus runtime library.

To add the DataJoiner Generic API entry points, you must execute the **mkrts** command that is provided by Micro Focus.

- Login as *root*.
- Execute **mkrts** with the arguments that are supplied in `/usr/lpp/djx_02_01_01/lib/db2mkrts.args`.

For example:

```
mkrts `cat /usr/lpp/djx_02_01_01/lib/db2mkrts.args`
```

Note: The `` character in the preceding command is the right prime character, which sometimes is referred to as the back-quote character.

Refer to the Micro Focus documentation for more information about the **mkrts** command and its other arguments that may be appropriate for your specific environment.

Adding DataJoiner Options after Initial Installation

When you install a software product or software product options on AIX 3.2, you can either apply (default) or commit the options. Applied options can be rejected or committed at a later date. Committed options are much more difficult to remove from the system. See "Deleting Options on AIX 3.2" on page 220 for more information.

Therefore, be sure to commit DataJoiner options only if you need to free disk space and you do not plan to remove the software from the system.

Attention: After an option is committed on AIX 3.2, you cannot reapply the option unless it is a newer version, release, or modification level or unless you force the installation by using the -F flag of `installp`. For example:

```
installp -F -d /dev/cd0 djx_02_01_01.djx_dba
```

On AIX 4.1, products are automatically committed. However, an `uninstall` option is available to delete installed products. See "Deleting Options on AIX 4.1" on page 221 for more information.

Multiple Instances

A DataJoiner server is referred to as an *instance* of the product. DataJoiner supports multiple instances on the same machine. To read a complete explanation, see "Multiple Instances" on page 226.

A Common Installation Problem for AIX 3.2

The DataJoiner options can be installed on systems that are running either AIX 3.2 or AIX 4.1. The operating system check is performed when the `djx_02_01_01.db2.rte` option is installed. Because of the way prerequisite failures are reported on AIX 3.2 systems, the `djx_02_01_01.db2.rte` option might fail to install and incorrectly report that an AIX 4.1 prerequisite is missing. When this problem occurs, it is usually caused by one of the following conditions:

- The operating system (AIX 3.2) is in an inconsistent state.
- A required AIX 3.2 PTF is missing.

To determine if there is an inconsistency in the AIX 3.2 operating system, enter the following command:

```
lppchk -v bos.obj
```

If this command reports any error conditions, correct the situation by installing the reported PTFs before you try to install the `djx_02_01_01.db2.rte` option again.

To check for the presence of a PTF that is required for a product, enter the following commands:

```
export ODMDIR=/usr/lib/objrepos  
odmget -q lpp_name=name product | grep PTFnumber
```

where *name* is the name of the product to which the PTF should be applied (for example, `bos.obj`) and `PTFnumber` is the number of a required PTF (PTFs have the form `U4XXXX`, where *X* indicates a digit from 0 to 9). Make sure that there are no spaces on either side of the equal sign (=).

If the required PTF is not found, you must install it before proceeding with the DataJoiner install.

Installation

This section contains instructions for installing DataJoiner and its components. Specific DataJoiner options are listed in “Smitty” on page 46. For information about installing the replication administration tool, see “Part 6. Replicating Data” on page 231.

There are four steps to install DataJoiner:

1. Restore the product files to the target machine.
2. Optional: Set up additional password encryption.
3. Link DataJoiner to data source client software.
4. Set up an instance of the product.

Each step is explained in this section.

Step 1: Restore the Product Files to the Target Machine

To restore the product files to the target machine, login as root, and use `installp` or `smitty` to transfer the files from the distribution medium.

Before you install DataJoiner, you might want to view the documentation that is included on the CD-ROM. To do so, you will first need to mount the CD-ROM using the following steps.



If you do not want to view documentation supplied on the CD-ROM, go directly to `Installp` or “Smitty” on page 46 for installation instructions.

1. Insert the CD into the CD-ROM drive.
2. Login as root.
3. Type the following on the AIX command line:

```
mkdir -p /usr/cdrom
```

where `/usr/cdrom` is the name of your root directory.
4. Allocate a CD-ROM file system by typing:

```
smitty storage
```
5. Select **File Systems**.
6. Select **Add / Change / Show / Delete File Systems**.
7. Select **CDROM File Systems**.
8. Select **Add a CDROM File System**.
9. Select the device name.
Requirement:
Device names for CD-ROM file systems must be unique. You may need to delete a previously defined CD-ROM file system or use another name for your directory if there is a conflict.
10. At the MOUNT POINT prompt, type:

```
/usr/cdrom
```
11. Exit from smitty.
12. Mount the CD-ROM file system by typing the following on the AIX command line:

```
smitty mountfs
```
13. Select (or enter) the FILE SYSTEM name (for example, `/dev/cd0`).
14. Select (or enter) the DIRECTORY name: `/usr/cdrom`.
15. Select (or enter) the TYPE of file system: `cdrfs`.
16. Set Mount as READ ONLY system to YES.
17. Select OK.
18. Exit from smitty and logout as root.

Use either `installp` or `smitty` to install DataJoiner. Instructions for using these utilities are provided in the following sections.



The AIX operating system provides detailed help for `installp` and `smitty`. Help for `installp` is available by entering `man installp`.

Installp

To install DataJoiner from a CD-ROM drive using `installp`, enter the following command:

```
installp -ag -d /dev/cd0 djx_02_01_01.db2.rte
```

This installs only the base DataJoiner product. To install other product options, you must enter their names separately. See “Smitty” for instructions on installing individual product components.

Smitty

The following instructions apply to AIX version 4.1. Previous versions of AIX might have renamed or reorganized the `smitty` panels. The overall process, however, should remain relatively similar.

To install using `smitty`:

1. Enter **smitty** from the command line.
2. Select **Software Installation and Maintenance**.
3. Select **Install and Update Software**.
4. Select **Install/Update Selectable Software (Custom Install)**.
5. Select **Install Software Products at Latest Level**.
6. Select **Install New Software Products at Latest Level**.
7. Press F4 to display a list of devices. Position the cursor on the CD-ROM drive where the DataJoiner CD-ROM is located and press Enter.
8. With the **SOFTWARE to Install** option highlighted, press F4 to display a list of software to install. Use the F7 key to select DataJoiner options:
 - To install the entire DataJoiner product and all its options, select `djx_02_01_01` (you will need to scroll right to see the abbreviated name). This automatically selects all indented options that are listed under the `djx_02_01_01` option. AIX 3.2.5 users need to select `djx_02_01_01.db2.rte`.
 - To install individual DataJoiner components, position the cursor to the left of each option, and press F7. See “Prerequisites for DataJoiner Options” on page 31 for information about individual options and their prerequisite requirements.
9. Press Enter and review your selections if necessary by scrolling right.
10. Press Enter again to begin the installation.

Step 2: (Optional) Set Up Additional Password Encryption

Perform this step only if you wrote your own password encryption and decryption functions that you want DataJoiner to use in addition to what is already provided.

To set up password encryption for installation:

1. Build an object file from your encryption source code. For example, if your source file is named `midEncrypt.C`, you must invoke a C++ compiler without link-editing with the following command:

```
x1C -c midEncrypt.C
```
2. As root, copy the resulting object file to the DataJoiner product directory for code libraries with the following command:

```
cp midEncrypt.o /usr/lpp/djx_02_01_01/lib
```
3. As root, make the library directory the current directory with the following command:

```
cd /usr/lpp/djx_02_01_01/lib
```
4. As root, add your object file to the DataJoiner library for user exits with the following command:

```
ar r libdjexits.a midEncrypt.o
```

This command replaces an existing object file. However, before you change it, the existing file does nothing except return the password encryption buffer unchanged.

DataJoiner is now ready to be link-edited with your encryption functions. Link-editing is covered in “Step 3: Link DataJoiner to Data Source Client Software”.

Not Planning to Link-Edit Your Data Sources?

There are rare situations in which you might not want to link-edit the data sources in “Step 3: Link DataJoiner to Data Source Client Software”. If you are not going to link-edit the data sources in the normal installation process, enable your additional encryption by following these steps:

1. As root, make the DataJoiner product directory for code libraries the current directory with the following command:

```
cd /usr/lpp/djx_02_01_01/lib
```
2. Proceed to “Running `djxlink.makefile`” on page 51, which is part of Step 3. You will have skipped the information about editing `djxlink.makefile`, which is provided in Step 3, because you are not going to link-edit data sources.

Step 3: Link DataJoiner to Data Source Client Software

To enable access to data source types, the DataJoiner server must be link-edited to the client libraries. The link-edit process creates a data access module for each data source with which the DataJoiner server will communicate. To build your data access modules, use the `djxlink.sh` script that is located in `/usr/lpp/djx_02_01_01/lib`. The `djxlink.sh` script creates a data access module for nearly all commonly-used data sources. The

| following example shows the typical results of running `djxlink.sh`:

```

Starting djxlink.sh on urchin at Wed Jun 3 09:58:15 PDT 1998

Warning:          Cannot prepare djxmssql Data Access Module for use
Problem:          DJX_ODBC_LIBRARY_PATH environment variable not set
Corrective action: Please set DJX_ODBC_LIBRARY_PATH environment variable to
                  the path where your ODBC driver is installed

Attempting to build dblib Data Access Module
  Successfully built dblib Data Access Module
Attempting to build ctlib Data Access Module
  Trying variation 1
  Trying variation 2
  Successfully built ctlib Data Access Module
Attempting to build eda Data Access Module
Warning:          Cannot prepare eda Data Access Module for use.
Problem:          EDA_HOME environment variable not correctly set
Corrective action: Please set EDA_HOME environment variable to the path where edalink
                  is installed and re-run djxlink.sh
Attempting to build informix Data Access Module
Warning:          Cannot prepare informix Data Access Module for use.
Problem:          INFORMIX_HOME environment variable not correctly set
Corrective action: Please set INFORMIX_HOME environment variable to the path where
                  Informix*Net is installed and re-run djxlink.sh
Attempting to build informix7 Data Access Module
  Trying variation 1
  Successfully built informix7 Data Access Module
Attempting to build sqlnet Data Access Module
  Trying variation 1
  Trying variation 2
  Trying variation 3
  Trying variation 4
  Trying variation 5
  Trying variation 6
  Trying variation 7
  Successfully built sqlnet Data Access Module
Attempting to build net8 Data Access Module
  Was not able to find link-edit parameters for net8 that would work
Attempting to build xaccess Data Access Module
  Successfully built xaccess Data Access Module

```

```

Results of djxlink
-----
djxmssql      Failure
dblib         Success
ctlib         Success
eda           Failure
informix      Failure
informix7     Success
sqlnet        Success
net8          Failure
xaccess       Success

```

```

Finished djxlink.sh on urchin at Wed Jun 3 09:58:59 PDT 1998

```

Figure 5. Example of output that results from running the djxlink.sh script ^{Installation} **49**

In most cases, running `djxlink.sh` is all you need to do to build your data access modules. However, if your library levels are different from those that are supported by `djxlink.sh`, you need to edit `djxlink.makefile`, also supplied in `/usr/lpp/djx_02_01_01/lib`, and run the makefile to link all needed data source types.

To use one or more of the data source types, you must ensure that the shell/environment variables needed to link that data access module are set correctly. For certain Oracle data types, you also must remove the comment `CHARACTER (#)` to allow DataJoiner to access it. Use `chmod` or `chown` commands to give you authorization to write the `djxlink.makefile` file when you edit it.

Consider the following information for each section of `djxlink.makefile` when editing the file:

For Oracle data sources:

- The shell/environment variables in this section need to be set only if you plan to use Oracle as a data source and have the appropriate Oracle licenses. The Oracle libraries you link-edit and the directories where the libraries reside might vary depending on the version of Oracle's client software (SQL*Net or Net8) that you are using. The link information is available in the Oracle installation log provided that the installer requested SQL*Net or Net8 be relinked during the install process.
- If you use SQL*Net, you must use the `sqlnet` data access module. Similarly, if you use Net8, you must use the `net8` data access module. Either option allows you to access both Oracle Version 7 and Oracle Version 8 data sources.
- If you are not sure which Oracle libraries to list in `djxlink.makefile` for the version and release of Oracle client software you have, please see the DataJoiner home page on the World Wide Web. This home page has a list of frequently asked questions (FAQ) that shows the libraries needed from some of the more common releases of both SQL*Net and Net8. The DataJoiner URL is:
<http://www.software.ibm.com/data/datajoiner/>

For Microsoft SQL Server data sources:

Support for Microsoft SQL Server is provided through Sybase Open Client (through the `dblib` API) or through Visigenic.

For Sybase data sources:

The shell/environment variables in this section need to be set if you plan to use Sybase as a data source and have the appropriate Sybase licenses. The Sybase libraries you link and the directories where the libraries reside might vary depending on which version of Sybase you are using. See the Sybase product documentation for Sybase link-edit requirements.

When running `djxlink.makefile`, modify either the `dblib` or `ctlib` section. `ctlib` is valid for Sybase 10 only. `dblib` is valid for Microsoft SQL Server, earlier versions of Sybase, and Sybase 10.

For example, to access Oracle 7.1.4, you change the Oracle section of `djxlink.makefile` from:

```
# Oracle 7.1.4
#sqlnet: liboracle.a oracle.exp
#           ld -o sqlnet \
#           -e _no_start \
#           -bE:oracle.exp \
#           -bI:libdb2e.exp \
#           -bM:SRE \
#           -K -lc -lld -lm \
#           -bI:$(ORACLE_HOME)/lib/mli.exp \
#           $(ORACLE_HOME)/lib/osntab.o \
#           -L$(ORACLE_HOME)/lib \
#           -lnlsrtl -lccic -lcore -lora -lcv6 -lsqlnet -lnetwork \
#           liboracle.a \
#           libdjexits.a
```

To:

```
# Oracle 7.1.4
sqlnet: liboracle.a oracle.exp
           ld -o sqlnet \
           -e _no_start \
           -bE:oracle.exp \
           -bI:libdb2e.exp \
           -bM:SRE \
           -K -lc -lld -lm \
           -bI:$(ORACLE_HOME)/lib/mli.exp \
           $(ORACLE_HOME)/lib/osntab.o \
           -L$(ORACLE_HOME)/lib \
           -lnlsrtl -lccic -lcore -lora -lcv6 -lsqlnet -lnetwork \
           liboracle.a \
           libdjexits.a
```

In this example, the comment symbol (`#`) is removed, thereby enabling the variables to be updated according to Oracle documentation.

Requirements:

- If you are using the `-b` flag, ensure the character that immediately follows is an upper-case 'I' and *not* a lower-case 'l' (the 'l' would cause an overwrite of the file, which is not what you want to have happen).
- For data sources that will be participating in two-phase commit transactions, you must modify `djxlink.makefile` to include XA library information. See "Chapter 15. Configuring for Two-Phase Commit" on page 169 for examples.

Running `djxlink.makefile`

1. If you are installing DataJoiner and have not already run `djxlink.makefile`, skip to step 2.

If you completed installation and returned to these instructions because you are adding or deleting a data source, stop all instances of DataJoiner by entering **db2istop** from `/usr/lpp/djx_02_01_01/instance`.

2. As root, make sure that you are in the library directory by entering the following command:

```
cd /usr/lpp/djx_02_01_01/lib
```

3. Check to see if you have enough free space in the /usr file system to run `djxlink.makefile` by entering:

```
df -I .
```

You will need less than 5 megabytes for each data access module that you are linking. If you do not have enough free space, add more.

4. Run the `djxlink.makefile`:

```
make -f djxlink.makefile nnnnnnnn
```

where *nnnnnnnn* is one of the following data sources:

- `ctlib`
- `dblib`
- `Eda`
- `Generic`
- `informix`
- `informix7`
- `net8`
- `mssqlodbc`
- `sqlnet`
- `Xaccess`

Note: Because you run `djxlink.makefile` from the `/usr/lpp/djx_02_01_01/lib` directory, you are enabling the data sources for all DataJoiner servers that will be defined on a given system.

5. Make `djxlink.makefile` readable by entering the following command:

```
chmod uog+r <data access module>
```

where *<data access module>* is the name of the data access module that you are linking.

6. If you are installing DataJoiner and have just run `djxlink.makefile` for the very first time, you can skip the remaining steps. Otherwise, if you have existing DataJoiner instances, run the **db2iupdt** command:

```
/usr/lpp/djx_02_01_01/instance/db2iupdt instanceName
```

where *instanceName* is the login name of the instance owner. You must run the **db2iupdt** command for each instance you have defined. If you do not run **db2iupdt** for an instance, that instance will not pick up the change from running `djxlink.makefile`.

7. If you stopped any existing instances in step 1, restart them by entering **db2start** at each instance.

When running `djxlink.makefile`, you might see several screens of warning messages scroll past your monitor. You can ignore the following messages:

- 0706-224 - Duplicate symbols
- 0706-223 - Replacement of local definitions
- 0706-221 - Replacement of import version
- 0711-327 - Entry point not found: `__start`
- 0706-406 - Symbols imported

A successful link is indicated by these messages:

- 0706-297 - No unresolved symbols detected
- 0706-476 - RC: Highest return code was 0

Step 4: Set up an Instance of the Product

You must create an instance of DataJoiner at install time to use the product. Because managing instances is an issue that you might face again if you decide to create additional instances in the future, issues pertaining to managing instances are covered in the next chapter.

To finish the installation of DataJoiner, complete the instructions in “Chapter 5. Setting up an Instance” on page 55.

Chapter 5. Setting up an Instance

A DataJoiner server is referred to as an *instance* of the product. You must create or migrate a DataJoiner instance to use the product. You might also want to create additional instances on the same AIX system in the future, for example:

- To differentiate between a production environment and a test environment
- To differentiate between groups of users to which you want to assign different levels of security
- To differentiate between certain groups of data sources that require different performance settings

To set up an instance of DataJoiner:

1. Create an instance of the product
2. Set environment variables for local clients
3. Set AIX processes
4. Start the instance
5. Create a DataJoiner database
6. Connect DataJoiner to data sources
7. Connect clients to DataJoiner

Each step is explained in this section.

Step 1: Create an Instance of the Product

You must perform this step as root. If you already identified an existing group and user, you can skip this step. If not, follow these steps to create a group, a user, and an instance:

1. Create a group, such as `djadm1`, to be the instance owner's group. You can create a group by issuing the **mkgroup** command or with `smitty`. To use the **mkgroup** command to create a group, enter:

```
mkgroup djadm1
```

Any user that belongs to this group has `sysadm` authority for this instance of DataJoiner, even if it is not the user's primary group.

To use `smitty` to create a group:

- a. Login as root user.
- b. Enter **smitty**.
- c. Select **Security & Users**.
- d. Select **Groups**.
- e. Select **Add a Group**.
- f. Type a unique group name and press Enter. For example, type `djadm1`.

- g. Press F3 twice to return to the **Security & Users** menu.
2. Create a user to contain the instance.

Use the **mkuser** command or smitty to create the user. The group name and login name must not begin with the string SQL SYS IBM PUBLIC GUESTS in uppercase or lowercase. Do not create a user name or instance name that is the same as an SQL reserved word.

To use the **mkuser** command, enter mkuser followed by the appropriate parameters. For example, enter:

```
mkuser pgrp=djadm1 groups=djadm1 home=/home/djinst1 djinst1
```

To use smitty to create a user and change the user password:

- a. Go to the **Security & Users** menu.
- b. Select **Users**.
- c. Select **Add a User**.
- d. Enter a unique user name. For example, djinst1.
- e. Enter the previously defined group name in the **Primary GROUP** field. In this example, use djadm1.

In addition to the rules that are imposed by the operating system for login names and groups, you must also adhere to the rules described in the following list:

- User names and group names must not start with:
 - a number (0 - 9)
 - ibm
 - sql
 - sys (allowed for group names)
- Additionally, group names must not start with:
 - admins
 - guests
 - local
 - users
- Both may contain only the following characters:
 - 0 through 9
 - a through z
 - @, #, or \$

- f. Enter the home directory. For example, type: /home/djinst1, and press Enter.
3. Change the user password:
 - a. Press F3 twice to return to the **Security & Users** menu.
 - b. Select **Passwords** from the **Security & Users** menu.
 - c. Select **Change a User's Password** from the **Passwords** menu.
 - d. Enter the user name you just created.
 - e. Enter a password for the new user.

- f. Enter the password a second time.
 - g. Press F10 to exit smitty.
4. Issue the **db2icrt** command to create a DataJoiner instance.

Issue the command by entering:

```
/usr/lpp/djx_02_01_01/instance/db2icrt instanceName -a Authtype -u UdfName
```

where:

- instanceName* Is the login name of the instance owner. In the examples in this section, it is djinst1.
- a Is an optional parameter that specifies the authentication type of the instance. Valid authentication types are SERVER, CLIENT, and DCS. If the -a parameter is not specified, the authentication type will default to SERVER.
- u Is an optional parameter that specifies the AIX user and group under which Fenced UDFs and DARIs will execute.

Restriction:

UdfName may not be **root** or **bin**.

If *UdfName* is unspecified, the user and group default to nobody:nobody; this means that there are no file access permissions defined for UDFs or DARIs.

You may set the *UdfName* later with the **db2iupdt** command, if you do not do it now.

Executing the **db2icrt** command creates the *INSTHOME*/sql1ib directory, where *INSTHOME* is the home directory of the instance owner. In our example, *INSTHOME* is /home/djinst1.

You should not place files or directories under *INSTHOME*/sql1ib except those that are created by DataJoiner or those that are specified in DataJoiner documentation. Keeping the *INSTHOME*/sql1ib directory clean prevents a loss of data if an instance is deleted.

Exception:

You can use the *INSTHOME*/sql1ib/function directory to store UDFs and DARI procedures.

Step 2: Set Environment Variables

This step describes how local clients gain access to DataJoiner and how environment variables are set for an instance. It is not required for remote clients.

DataJoiner builds two script files when the instance is created and puts the files under the instance owner's home directory, in a directory named sql1ib. These scripts are db2profile for the Bourne and Korn shells and db2cshrc for the C shell.

You use these files to set up the database environments. These scripts can be invoked from a local user's and the instance's .profile or .login file. It is important to use only the script file that is appropriate to the shell that you use in your environment. Remember to either log out and log in again or execute the .profile to invoke the db2profile.

These scripts contain statements to perform the following tasks:

1. Update the PATH environment variable by adding the following directories:

```
INSTHOME/sql11ib/bin
INSTHOME/sql11ib/adm
INSTHOME/sql11ib/misc
```

where INSTHOME is the home directory of the instance owner.

2. Set the DataJoiner environment variables. Both local clients and DataJoiner instances use db2profile. Of the environment variables that are listed here, DB2INSTANCE is the only one a local client must set or change to access a DataJoiner instance. A local client can accept the default values for all other environment variables. Some of these variables are only used by DataJoiner instances. You can either accept the defaults or change the values based on your needs. The environment variables are:

DB2BQTIME Default = 1 second. Minimum = 1 second.

Specifies the amount of time the DataJoiner command line processor front end sleeps before checking if the back end process is active and establishing a connection.

DB2BQTRY Default = 60 retries. Minimum = 0 retries.

Specifies the number of times the DataJoiner command line processor front end process tries to determine whether the back end process is already active. It works in conjunction with DB2BQTIME.

DB2CHKPTR Default = null (OFF). Any Value = (ON).

Selectively turns pointer checking ON for debugging. It should be left OFF for normal operation.

DB2COMM Default = null. Values: APPC, IPXSPX, NetBIOS, TCPIP, NONE, or others.

Specifies whether or not the communication protocol is enabled when the DataJoiner server is started. This environment variable is not used by a local client. It is discussed more completely in "Chapter 20. Controlling Communication Support Initialization" on page 205.

This variable can only be set after the database manager configuration file has been updated as described in "Chapter 20. Controlling Communication Support Initialization" on page 205. If the variable is set prior to this, the database manager will not start.

DB2DBDFT Default = SAMPLE.

Specifies the database alias name of the DataJoiner database to which applications are implicitly connected when started.

DJXCOMM Default = null (OFF). Value: *data-access-module*.

Specifies whether or not a data access module is preloaded into the DataJoiner instance. By default, DataJoiner loads a data access module each time a data source is accessed, and the data access module is not loaded. DJXCOMM tells DataJoiner to load a data access module when DataJoiner is started. Preloading data access modules can improve performance since the data access module only loaded once between the time DataJoiner is started and stopped. Possible values include drda, db2ra, informix, informix7, djxclassic, and others. These values are the same as those that are found in the SERVER_PROTOCOL column of SYSCAT.SERVERS. See “Chapter 6. Configuring Access to Data Sources” on page 69 for more information about data access modules.

Multiple data access modules can be loaded using DJXCOMM.

The syntax for DJXCOMM is:

```
►►—export—DJXCOMM="—data-access-module—"—————►►
```

DJX_NR_CONFIG Default = null, Value: name of the Classic Connect config file.

This environment variable is only used by remote clients and specifies the complete path and file name of the configuration file for the DataJoiner Classic Connect interface. If set and the file exists, support for Classic Connect is enabled. If null, not set, or DataJoiner cannot find the file, the Classic Connect interface cannot be used.

The Classic Connect interface is used to access data sources that are defined to DataJoiner Classic Connect for MVS. A sample setting is:

```
DJX_NR_CONFIG=instancehomedir/sql1lib/djxclassic.cfg
```

See the *DataJoiner Classic Connect Planning, Installation, and Configuration Guide* for more information.

DB2INSTANCE Default = instance_name.

Specifies the instance that is active by default.

DB2IQTIME Default = 5 seconds. Minimum = 1 second.

Specifies the amount of time the DataJoiner command line processor back end process waits on the input queue for the front end process to pass commands.

DB2RQTIME Default = 5 seconds. Minimum = 1 second.

Specifies the amount of time the DataJoiner command line processor waits for a request from the front end process.

An instance owner or sysadm user can customize the db2profile or db2cshrc script files for all users of an instance. Alternatively, each user can copy and customize these scripts to a specific environment.

There are several methods to invoke one of these scripts. For a Bourne or Korn shell, choose one of the following techniques:

- Concatenate db2profile with the user's .profile by entering:
cat /home/djinst1/sql1lib/db2profile >> /home/djinst1/.profile
- Execute db2profile from the user's .profile by typing a new line into the .profile:

- /home/djinst1/sql1lib/db2profile
- Manually execute the script when connection to a database is needed by entering:
 - /home/djinst1/sql1lib/db2profile
- Develop a customized procedure for users. If you choose this option, be sure to run **db2 terminate** before switching instances.

For a C shell, choose one of the following techniques:

- Concatenate db2cshrc with the user's .login or .cshrc by typing:


```
cat /home/djinst1/sql1lib/db2cshrc >> /home/djinst1/.login
```

or

```
cat /home/djinst1/sql1lib/db2cshrc >> /home/djinst1/.cshrc
```
- Execute the **db2cshrc** command from the user's .login by typing a new line into the .login:


```
source /home/djinst1/sql1lib/db2cshrc
```
- Manually execute the script when connection to a database is needed by entering:


```
source /home/djinst1/sql1lib/db2cshrc
```
- Develop a customized procedure for users. If you choose this option, be sure to run **db2 terminate** before switching instances.

Step 3: Set AIX Processes

AIX provides mechanisms that can restrict the number of processes that can execute under a given owner id. This limit is controlled through the AIX parameter maxuproc, which has a default setting of 40. The default limit of 40 might not be sufficient if the system is required to support many databases and connections.

The following error occurs on connect to the DataJoiner database if the maxuproc value is exceeded:

```
SQL1224N A database agent could not be started to service a request
or was terminated as a result of a database system shutdown or a force
command.
```

To see the current definition of maxuproc, enter:

```
lsattr -E -l sys0
```

To view the number of processes that are currently executing on the system, enter:

```
ps -ef|grep instdj1|wc -l
```

where *instdj1* represents your DataJoiner instance owner's id.

To change the value of maxuproc, enter:

```
chdev -l sys0 -a maxuproc='nn'
```

where *nn* is the new integer value of the parameter.

Step 4: Start the Instance

Start the DataJoiner instance by entering:

```
db2start
```

Step 5: Create a DataJoiner Database

DataJoiner uses the catalog from a local database to maintain information about the DataJoiner heterogeneous environment. Therefore, you must create a DataJoiner database.

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

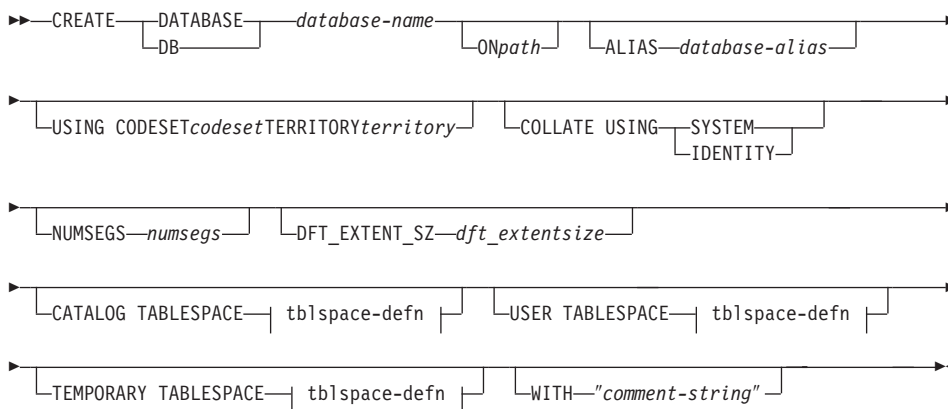


Figure 6. CREATE DATABASE Command Syntax

Notes:

1. The code set and territory values that are specified must be a valid combination. For a list of valid combinations, see the *DB2 for AIX Installation and Operation Guide*.
2. 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 as documented in the *DB2 Command and API 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 is invoking the command. The application code page is derived from the active environment when the database connection is made.

In UNIX-based environments, the active environment is determined from the environment variables that are associated with the locale, which include information about language, territory, and code set.

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 the discussion of National Language Support in “Appendix C. National Language Support” on page 325.

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, performance and data usage 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.

Your AIX file system may have limitations on the size of the files and directories. To overcome this limitation, DataJoiner lets you create databases that span multiple file systems by distributing databases and tables across multiple disk segments.

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 are 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 “Step 2: Set Environment Variables” on page 57. If so, set DB2DBDFT to the name of the newly created database.

Creating the Database

From the login for the DataJoiner instance that you created in step 1, issue the **CREATE DATABASE** command. For example, type:

```
db2 create database djdb1 on /db1and2
```

Spatial Extender users:

If you plan to run the sample application that is provided with the Spatial Extender, name your database “sample.” Refer to the *IBM DB2 Spatial Extender Administration Guide and Reference* for more information.

Step 6: Connect DataJoiner to Data Sources

This step is covered fully in “Part 3. Connecting DataJoiner to Data Sources” on page 67. After reading and following the instructions on Part 3., proceed to Step 7 below.

Step 7: Connect Clients to DataJoiner

Follow the instructions in “Part 4. Connecting Clients to DataJoiner” on page 173 to finish setting up this instance.

Step 8: Enable a DataJoiner Database for Spatial Data

If you plan to work with spatial (GIS) data on a DataJoiner database, you must perform this step. See the *IBM DB2 Spatial Extender Administration Guide and Reference* for instructions.

Part 3. Connecting DataJoiner to Data Sources

Chapter 6. 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.

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 101
DataJoiner V1.2 and V2	IBM DB2RA or DRDA	page 101 (for DB2RA) or page 73 (for DRDA)
DB2 Universal Database V5	IBM DB2RA or DRDA	page 101 (for DB2RA) or page 73 (for DRDA)
DB2 for OS/390, DB2/MVS, DB2 for OS/400, SQL/DS and DB2 for CS V2 (See Table Notes)	DRDA	page 73
DB2 for CS (except DB2 for OS/2 V1 and DB2 Parallel Edition)	IBM DB2RA	page 101
DB2 Parallel Edition	IBM DB2RA or DRDA	page 101 (for DB2RA) or page 73 (for DRDA)
IMS	Classic Connect	<i>the DataJoiner Classic Connect Planning, Installation, and Configuration Guide</i>
Informix OnLine V5	informix data access module	page 131
Informix OnLine V7.1 (non-GLS, -XPS, and -SE versions only)	informix7 data access module	page 137
Microsoft SQL Server	Sybase Open Client or Visigenic	page 143 (for Sybase Open Client) or 149 (for Visigenic)
ODBC/XOpen CLI	Generic Data Access Module	page 165
Oracle	Oracle SQL*Net or Net8	page 155
Sybase	Sybase Open Client	page 143

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

VSAM	Classic Connect	the <i>DataJoiner Classic Connect Planning, Installation, and Configuration Guide</i>
Other relational or non-relational data sources	CrossAccess	page 159

Table Notes:

- Although you can use DRDA to access DB2 for CS V2, IBM DB2RA is the recommended access method for DB2 for CS.
- DRDA application server data sources that will be participating in two-phase commit transactions must use the DB2RA protocol; see “Chapter 15. Configuring for Two-Phase Commit” on page 169.
- 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.

Link-Editing Data Access Modules

To access many data sources, DataJoiner libraries must be link-edited with data source client libraries through the `djxlink.sh` script. For most data sources, the `djxlink.sh` script builds an executable file that is called a data access module. DataJoiner communicates with data sources through the use of data access modules. The data access module name is used as the protocol name in the `CREATE SERVER MAPPING` statement when you configure DataJoiner for a data source.

Exception:

DB2 data sources do not need to be link-edited with DataJoiner libraries.

DataJoiner is designed to link-edit data access modules directly to remote data sources client code. See “Step 3: Link DataJoiner to Data Source Client Software” on page 47 for more information.

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 “Step 2: Set Environment Variables” on page 57 for more information.

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 95 and "Accessing DB2RA Data Sources Outside the Heterogeneous Environment" on page 128.

Chapter 7. Accessing DRDA Data Sources

There are two ways to access data sources through the DataJoiner DRDA Application Requester function:

- DataJoiner provides access through the use of nicknames and the pass-through facility. This process is explained in “Accessing DRDA Data Sources using Nicknames and Pass-through”.
- DataJoiner also provides access for applications that use IBM’s Distributed Database Connection Services (DDCS). This process is explained in “Accessing DRDA Data Sources using DataJoiner DDCS” on page 95.

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 “Chapter 15. Configuring for Two-Phase Commit” on page 169 .

The configuration steps that are required to access a DRDA 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 and user information
4. Test connections

Figure 7 on page 74 is an overview that shows how DataJoiner’s DRDA application requester accesses a data source by using the catalog view, SYSCAT.SERVERS. Figure 7 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.

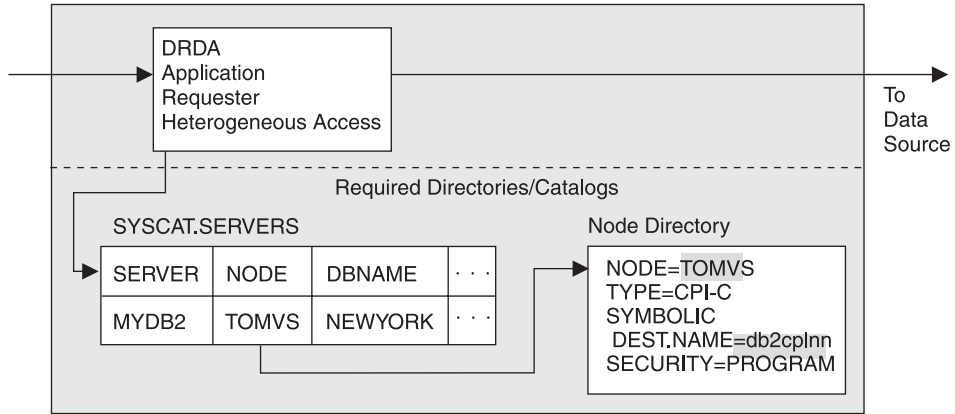


Figure 7. DataJoiner Application Requester Heterogeneous Access

The remainder of this section contains detailed instructions for the steps that were just discussed. It will be helpful to fill in the planning template that is provided as you configure your environment. These templates are for DB2 for MVS; other data sources are similar.

DataJoiner

SYSCAT.SERVERS

SERVER	NODE	DBNAME	...
<input type="text"/>	<input type="text"/>	<input type="text"/>	...

SYSCAT.TABLES

Nickname
 Table is at

Node Directory

Node=
 CPI-C
 Sym. Dest. name=
 Security Type=

SNA Server/6000

Control Point Profile
 Network name=
 XID Node ID
 LU 6.2 Local LU Profile
 Local LName=
 Dependent LU? no
 Partner LU 6.2 Location Profile
 Fully qualified partner LName=
 Local LName=
 Link Station Profile name=
 Link Station Profile
 Profile name=
 Use Control Point XID? yes
 Remote Link Address=
 LU 6.2 Mode Profile
 Mode name=
 LU 6.2 Side Information Profile
 Profile name=
 Local LU=
 Fully qualified partner LName=
 RTPN=
 RTPN in hex? yes

**DB2 for MVS
Data Source**

VTAM owning RISC

NETID
 PU
 IDBLK=
 IDNUM=
 LU
 LOCADDR=0
 MODEENT

NCP - NTRI

TIC Address
 LOCADD=

VTAM owning host

NETID
 APPL
 MODEENT

**DB2 Communications
Database (DB)**

SYSIBM.SYSLUNAMES (LUNAME)

 SYSIBM.SYSUSERNAMES (LUNAME)

DB2 BSDS (Bootstrap Data Set)

LOCATION=
 LUNAME=

Figure 8. DRDA Data Source Planning Template

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.

- To configure TCP/IP, see “Configuring TCP/IP for DRDA3 Data Sources”.
- To configure SNA LU 6.2, see “Configuring SNA Server/6000 for DRDA Data Sources” on page 78.

Configuring TCP/IP for DRDA3 Data Sources

This section shows you how to establish a connection from DataJoiner to a DRDA3 data source through TCP/IP. This section can be used as a guide for configuring any DRDA3 data source that is accessed through 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.

The values that are used in this section are in Figure 18 on page 102. We also provide a blank template in Figure 19 on page 104. 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.

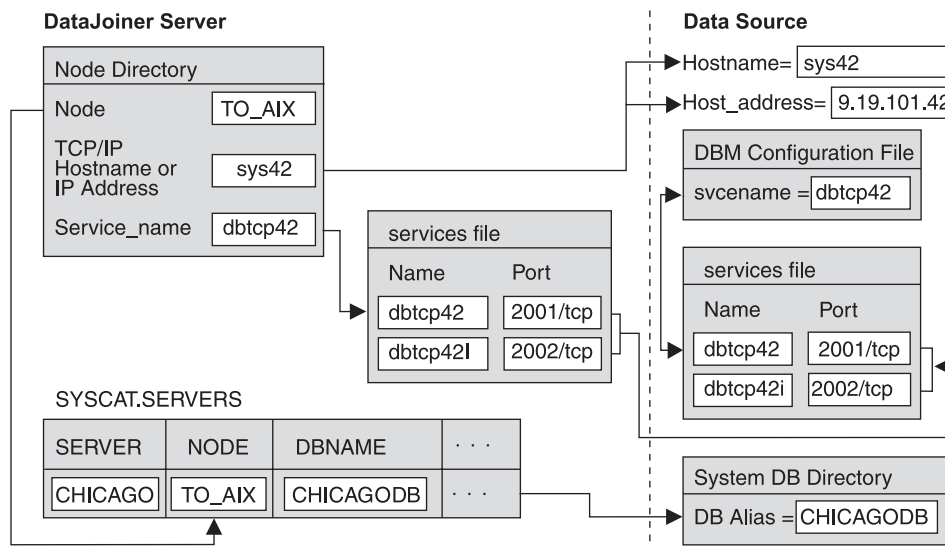


Figure 9. DRDA Data Source through TCP/IP

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 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 numbers for the data source as it is stored in the `/etc/services` file on the data source's system. In the example, the entry would be:

```
dbtcp42    3700/tcp      # Data source primary connect port
dbtcp42i  3701/tcp      # Data source interrupt port
```

Note: 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. Define port numbers. 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, you will need to define them using the following steps.
 - a. Find the data source's port numbers in its system's `/etc/services` file.

- b. Use one of the following methods to define the data source's port numbers to DataJoiner.
 - Update DataJoiner's /etc/services file with the data source's port numbers. In the example, the entries would be:


```
dbtcp42    3700/tcp      # Data source primary connect port
dbtcp42i   3701/tcp      # Data source interrupt port
```
 - Specify the port numbers on the server parameter of the **CATALOG TCPIP NODE** command, which is discussed in "Step 2: Update the Node Directory" on page 92.

2. Synchronize the /etc/services file and the inetd daemon by issuing the following AIX commands:

```
inetimp
refresh -s inetd
```

This step is not required for AIX V4 or higher.

3. As root user, update /etc/hosts for the data source workstation, if necessary. Whether you must update /etc/hosts 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 /etc/hosts. Otherwise, you need an entry in /etc/hosts for the remote host name. See your network administrator to learn how your network is configured.



When you have finished setting up TCP/IP network communications, continue the configuration process. If you are configuring for access within the heterogeneous environment, go to "Step 2: Update the Node Directory" on page 92. If you are configuring for access outside the heterogeneous environment, go to "Step 2: Update the Node Directory" on page 96.

Configuring SNA Server/6000 for DRDA Data Sources

To configure SNA resources you must configure SNA profiles. Use the smitty interface to configure profiles. This section shows one method of configuring the profiles. Refer to *AIX SNA Server/6000 Configuration Reference SC31-7014* for detailed information. To access a DRDA data source, the following items must be configured:

- Physical connections
- Links
- Control point profile
- LU 6.2 sessions

The samples that follow assume you are using SNA Server/6000 V2. To access the SNA profiles with smitty:

1. Enter **smitty**.
2. Select **Communications Applications and Services**.

3. Select **SNA Server/6000**.
4. Select **Configure SNA Profiles**.
5. Select **Advanced Configuration**. From there, you can define links, sessions, control points, and profiles.

Physical Connections

SNA Server/6000 supports five communication adapters. The examples in this section are for configuring a token-ring adapter.

Links

Links depend on the type of physical connection to your system. SNA Server/6000 V2 supports multiple call and listen link stations on one token-ring adapter. For this reason you must ensure that you have one of each of the following profiles:

- Token-Ring SNA DLC profile (usually one per workstation)
- Token-Ring Link Station profile (usually one per link to each data source host)

The steps that follow assume that a Token-Ring SNA DLC profile and a Token-Ring Link Station profile must be created. To create a Token-Ring SNA DLC profile:

1. From the Advanced Configuration menu that you accessed previously, select **Links**.
2. Select **Token-Ring**.
3. Select **Token-Ring SNA DLC**.
4. Select **Add a Profile**.

The **Change/Show Token-Ring SNA DLC Profile** window opens (Figure 10 on page 80).

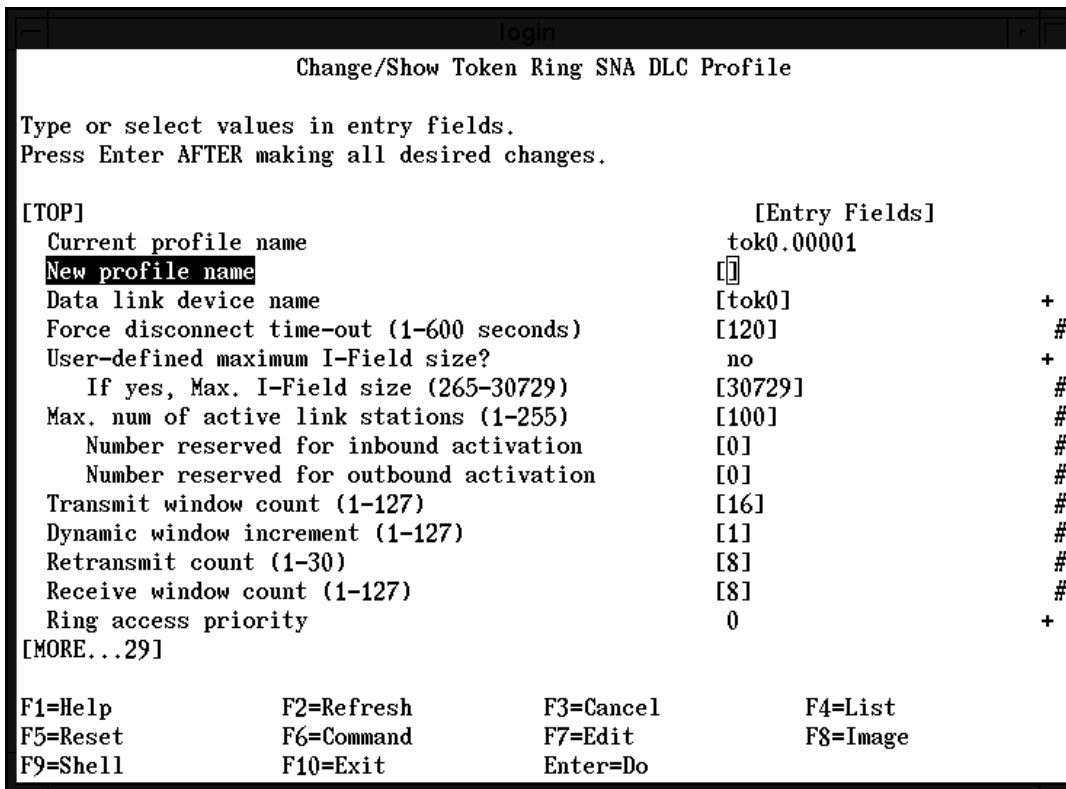


Figure 10. Token-Ring SNA DLC Profile

When filling out the **Add Token-Ring SNA DLC Profile** menu, consider the following:

- **Data link device name** is the RISC System/6000's token-ring card that is associated with this link. The default is tok0. If the RISC System/6000 has only one Token-Ring adapter, this field is tok0.
- The default **Current Profile Name** is tok0.00001 for token-ring adapter card 0.
- This configuration assumes that **Dynamic link stations supported?** is set to the default value of yes. Note that you must scroll to see this field.
- The remaining fields can use the default values provided by SNA Server/6000.
- Press Enter after filling out the profile, and check the output screen for syntax errors.
- Press F3 to return to the **Token-Ring physical** unit menu.

Now you must select **Add a Profile** from the **Token-Ring Link Station** menu choice. At the link station profile screen, shown in Figure 11 on page 81, perform the following steps:

```

login
Change/Show Token Ring Link Station Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...1]                                [Entry Fields]
New profile name                            []
Use Control Point's XID node ID?            yes          +
  If no, XID node ID                        [*]
* SNA DLC Profile name                       [tok0.00001]    +
Stop link station on inactivity?            no            +
  If yes, Inactivity time-out (0-10 minutes) [0]           #
LU address registration?                     no            +
  If yes,
  If yes, LU Address Registration Profile name []          +
Trace link?                                  no            +
  If yes, Trace size                         long          +

Adjacent Node Address Parameters
Access routing                              link_address    +
[MORE...39]

F1=Help          F2=Refresh          F3=Cancel          F4=List
F5=Reset         F6=Command          F7=Edit           F8=Image
F9=Shell         F10=Exit            Enter=Do

```

Figure 11. Token-Ring Link Station Profile

1. Provide a descriptive profile name for the link station profile. Try to provide a name that defines the host on which the data source is located.
2. The XID node ID can be coded in the Control Point profile or the Link Station profile. The default value for **Use Control Point's XID node ID?** is yes.
3. Type the **SNA DLC Profile name** that you previously used (that is, tok0.00001).
4. Set the value of the **Adjacent Node Address Parameter** to link_address.
5. The **Remote link address** field is the universal or locally administered token-ring address of the destination host. In this example, the value is 400009451007, but you must scroll the screen to see the field. For a VTAM host, this value will be the TIC address of the 3745 communications controller, the 3172 token-ring adapter address, or the 3174 token ring address.
 If a 3745 communications controller is being used, the locally administered address can be found in the NCP listing that defines the TIC. Look for the LINE statement for the TIC, and use the LOCADD= field. This value should be a 12-position hex number that probably starts with 4000.
6. Use the default values provided by SNA Server/6000 in the remaining fields.

7. Press Enter after filling out the profile, and check the output screen for syntax errors.
8. Press F3 to return to the **Advanced Configuration** menu.

Control Point Profile

To configure a control point profile:

- From the **Advanced Configuration** menu, select **Control Point**.
- Select **Change/Show a Profile**.

The **Change/Show Control Point Profile** window opens.

In most cases, the profile is already created for 3270 LU Type 2 connections and can be used again. If the profile does not exist, choose **Add a Profile**.

```

login
Change/Show Control Point Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Profile name                    node_cp
  XID node ID                     [071b8004]
  Network name                     [USIBMST]
  Control Point (CP) name          [DJCP]
  Control Point alias              [DJCP]
  Control Point type               appn_end_node      +
  Maximum number of cached routing trees [500]          ##
  Maximum number of nodes in the TRS database [500]          ##
  Route addition resistance         [128]             ##

  Comments                         []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 12. Control Point Profile

In **Change/Show a Profile**:

- The default profile name is node_cp.

- The XID node ID can be coded in the Control Point Profile or the Link Station Profile.
- The **Control Point type** field can be appn_end_node (EN) or appn_network_node (NN). Type appn_end_node unless in an APPN network. If in an APPN network, type appn_network_node (NN).
- Use default values provided by SNA Server/6000 in the remaining fields.
- Press Enter after filling out the profile and check the output screen for syntax errors.
- Press F3 to return to the **Advanced Configuration** menu.

LU 6.2 Sessions

After physical unit (PU) definitions are complete, you must define LU 6.2 definitions. LU 6.2 sessions require the following profiles be created:

- Local LU
- Partner LU 6.2 Location profile
- Mode
- LU 6.2 Side information

Local LU

To define Local LU profiles:

1. From the **Advanced Configuration** menu, select **Sessions**.
2. Select **LU 6.2**.
3. Select **LU 6.2 Local LU**.
4. Select **Add a Profile..**

The **Change/Show LU 6.2 Local LU Profile** window opens.

login

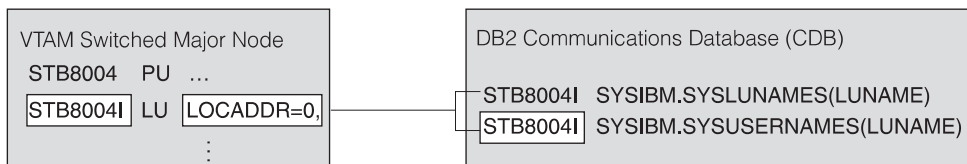
Change/Show LU 6.2 Local LU Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Current profile name	ludj1	
New profile name	[]	
Local LU name	[STB8004I]	
Local LU alias	[STB8004I]	
Local LU is dependent?	no	+
If yes,		
Local LU address (1-255)	[]	#
System services control point (SSCP) ID (*, 0-65535)	[*]	
Link Station Profile name	[]	+
Conversation Security Access List Profile name	[]	
Comments	[]	

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 13. LU 6.2 Local LU Profile



5. Even though this is an independent LU, there will be a local VTAM host which has the PU and LU SNA resource definitions. The **Local LU name** field comes from the local VTAM host that has the PU and LU definitions. From either the NCP or the Switched Major Node definitions, this is the name on the LU statement where LOCADDR = 0. It is a unique value that is assigned to this workstation.
6. The communications database (CDB) on DB2 might have to be updated for the local LU that is defined in this profile (STB8004I in this example). You can use default values in the CDB for this LU, or you can add a row that specifically

authorizes this LU. Refer to the *DRDA Connectivity Guide* (SC26-4783) for more information on setting up SYSIBM.SYSLUNAMES and SYSIBM.SYSUSERNAMES.

7. Be sure the **Local LU is dependent?** field is set to No. No is the default.
8. Leave the remaining fields alone.
9. Press Enter after filling out the profile and check the output screen for syntax errors.
10. Press F3 to return to the LU 6.2 menu.

Partner LU 6.2 Location Profile

Next, configure the Partner LU 6.2 location from the **LU 6.2** menu:

1. Select **Partner LU 6.2 Location** (not LU 6.2 Partner LU).
2. Choose **Add a Profile**.

login

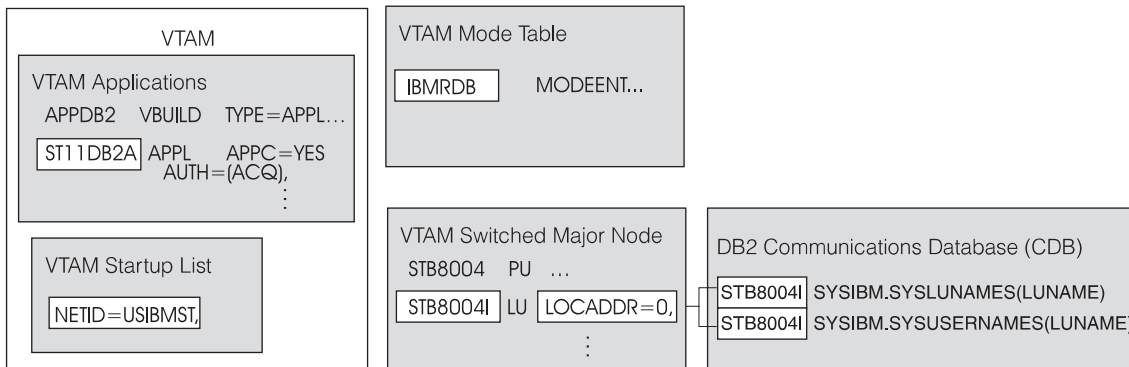
Change/Show Partner LU 6.2 Location Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Current profile name	drdadbdj1	
New profile name	[]	
Fully qualified partner LU name	[USIBMST.ST11DB2A]	
Partner LU location method	link_station	+
If owning_cp,		
Fully qualified owning Control Point (CP) name	[]	
Local node is network server for LEN node?	no	+
Fully qualified network node server name	[]	
If link_station,		
Local LU name	[STBS004I]	
Link Station Profile name	[djtr]	+
Comments	[]	

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 14. Partner LU 6.2 Location Profile



- The Partner LU 6.2 Location Profile points to the local LU, the partner LU, and defines the partner LU location method. In this example, `link_station` is used to locate the partner LU.
- A separate Partner LU 6.2 profile is needed for each DRDA application server subsystem to which DataJoiner is to connect. The Link Station profile points back to the same local LU profile.
- For the new profile name, use a descriptive name. Consider using a name that defines the DRDA application server subsystem on the host.
- The Network name in the partner LU name is the network on which the DRDA application server's LU is defined. This network can be the same network as the RISC System or another network. The value is in the remote VTAM host's startup list. In this example, `NETID = USIBMST`.
- The Partner LU name is the LU that defines the application server subsystem. On the VTAM host that contains the application server subsystem, use the name on the APPL statement that defines the DRDA application server program. In this example, `ST11DB2A` is the name on the APPL statement.
- The Partner LU location method selected for this example is `link_station`.
- Use the local LU that is defined in Figure 13 on page 84 for the **Local LU name** field.
- The Link Station Profile name field points to the token ring link station profile, defined in Figure 11 on page 81, used to connect to the remote host.
- Accept default values for the remaining fields.
- Press Enter after filling out the profile and check the output screen for syntax errors.
- Press F3 to return to the **LU 6.2** menu.
- The **Local LU name** is the local LU defined in the Local LU profile that was described in the previous step.

Mode

Modes, such as IBMRDB, must be unique, but they can be shared by multiple applications. The same mode can be used for remote DataJoiner clients and remote data sources. This example assumes that the mode already exists. To access the LU 6.2 mode profile:

1. Select **LU 6.2 Mode**.
2. Select **Change/Show a Profile**.
3. Press F4 to see a list of all existing mode profiles. Highlight IBMRDB and press Enter.

```
login
Change/Show LU 6.2 Mode Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Current profile name          [Entry Fields]
New profile name             [ ]
Mode name                    [IBMRDB]
Maximum number of sessions (1-5000) [32]
Minimum contention winners (0-5000) [16]
Minimum contention losers (0-5000) [16]
Auto activate limit (0-500) [0]
Upper bound for adaptive receive pacing window [16]
Receive pacing window (0-63) [3]
Maximum RU size (128,...,32768: multiples of 32) [2048]
Minimum RU size (128,...,32768: multiples of 32) [256]
Class of Service (COS) name [ #CONNECT]
Comments                     [ ]

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command     F7=Edit      F8=Image
F9=Shell     F10=Exit       Enter=Do
```

Figure 15. Mode Profile

VTAM Mode Table	
IBMRDB	MODEENT ...

Keep the following in mind:

- The **Mode name** field must match the mode name that is specified in the Side Information Profile. The Side Information Profile is described in Step 2: Update the Node Directory.
The **MODE name** field contains the mode name that is to be sent during the LU-to-LU bind. Both the local VTAM host and the Application Server VTAM host must have a mode table entry with the same MODEENT name. The mode name must be entered in uppercase.
The profile name and the Mode name do not need to be the same.
- The mode profile defines the maximum number of concurrent sessions that are allowed between DataJoiner and an application server.
The **Maximum number of sessions** field determines how many concurrent client connections are to be allowed to a particular application server. If DataJoiner has remote clients, ensure that the minimum number of contention winners is greater than 0. Together, the winners and losers cannot exceed the maximum number of sessions.
- The pacing fields set how many LU frames are to be sent or received before an acknowledgment is required.
- The RU size fields set the largest and smallest request/response unit size that can be handled by an application. The suggested maximum RU size is 4096 bytes. The suggested minimum RU size is 1024 bytes. The maximum RU size allowed for SNA Server/6000 is 32,768 bytes. The minimum RU Size allowed for SNA Server/6000 is 128. The RU size must be specified in multiples of 32 bytes.
- The suggested class of service name is #CONNECT. It is the default.
- Press Enter after filling out the profile, and check the output screen for syntax errors.
- Press F3 to return to the **LU 6.2** menu.

LU 6.2 Side Information

The LU 6.2 Side Information profile is pointed to by the Node directory. The profile points to the partner LU (DRDA AS). You must create a Side Information profile for each DRDA application server subsystem. From the **LU 6.2** screen:

1. Select **LU 6.2 Side Information**.
 2. Select **Add a Profile**.
- The **Change/Show LU 6.2 Side Information Profile** window opens.

```

Change/Show LU 6.2 Side Information Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

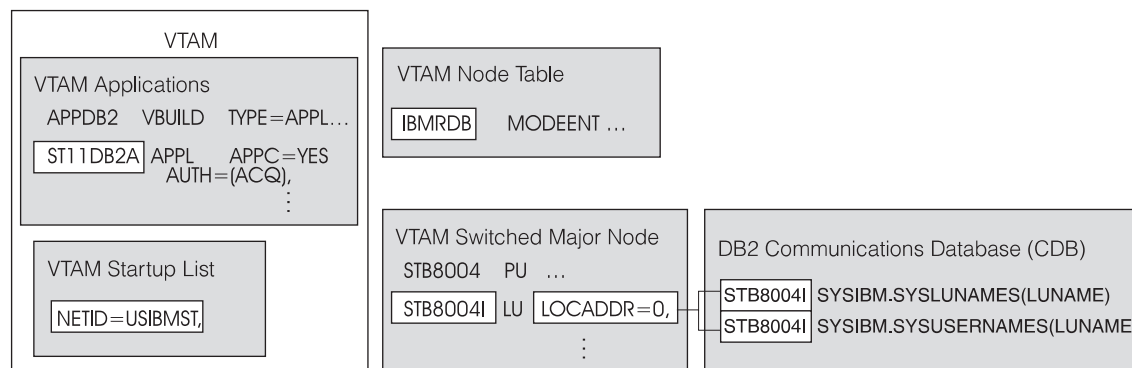
Current profile name          [Entry Fields]
New profile name             db2cpidj
Local LU or Control Point alias [STB8004I] +
Provide only one of the following:
  Partner LU alias           [] +
  Fully qualified partner LU name [USIBMST.ST11DB2A]
Mode name                    [IBMRDB] +
Remote transaction program name (RTPN) [07F6C4C2]
RTPN in hexadecimal?         yes +

Comments                      []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 16. LU 6.2 Side Information Profile



3. Choose a descriptive profile name. You will use it in “Step 2: Update the Node Directory” on page 92.

4. The Local LU name field comes from the local VTAM host that has the PU and LU definitions. From either the NCP or the Switched Major Node definitions, this is the name on the LU statement. This name, STB8004I in this example, was defined in the LU 6.2 Local LU profile and optionally in the CDB in an earlier step, as shown in Figure 13 on page 84.
5. The fully qualified partner LU name consists of a Networkid.PartnerLUname which, in this example, is USIBMST.ST11DB2A. The **NETWORK name** is the network where the DRDA application server's LU is defined. This network can be the same network as the RISC System or another network. The value is in the remote VTAM host's startup list. In this example, NETID = USIBMST. The **Partner LU name** is the LU that defines the application server subsystem. On the VTAM host that contains the application server subsystem, use the name on the APPL statement that defines the DRDA application server program, which is ST11DB2A in this example.
6. The **Mode name** is the name of the mode that is passed during LU bind processing. The name must equal a mode name in a LU 6.2 Mode profile and must be entered in uppercase. In this example it is IBMRDB, which is taken from Figure 15 on page 87.
7. The **Remote Transaction Program name** field is the name of the application server transaction program. This field must contain the value X'07F6C4C2' in a DB2 for OS/390 environment.
For DB2 for OS/400, the value is also X'07F6C4C2'. In SQL/DS, the value is defined by the user. For details about configuring these environments, refer to the *DRDA Connectivity Guide (SC26-4783)*.
8. Set the **RPTN in hexadecimal?** field to yes. This value is not the default.
9. Press Enter after filling out the profile and check the output for syntax errors.
10. Press F3 to return to the **Advanced Configuration** menu.

SNA System Defaults

The SNA Node Profile is used to set global limits on the number of concurrent sessions and conversations. To set SNA system defaults:

1. From the **Advanced Configuration** menu, select **SNA System Defaults**.
2. Select **Change/Show a Profile**.

This displays the **Change/Show SNA Node Profile** screen, as shown in Figure 17 on page 91 .

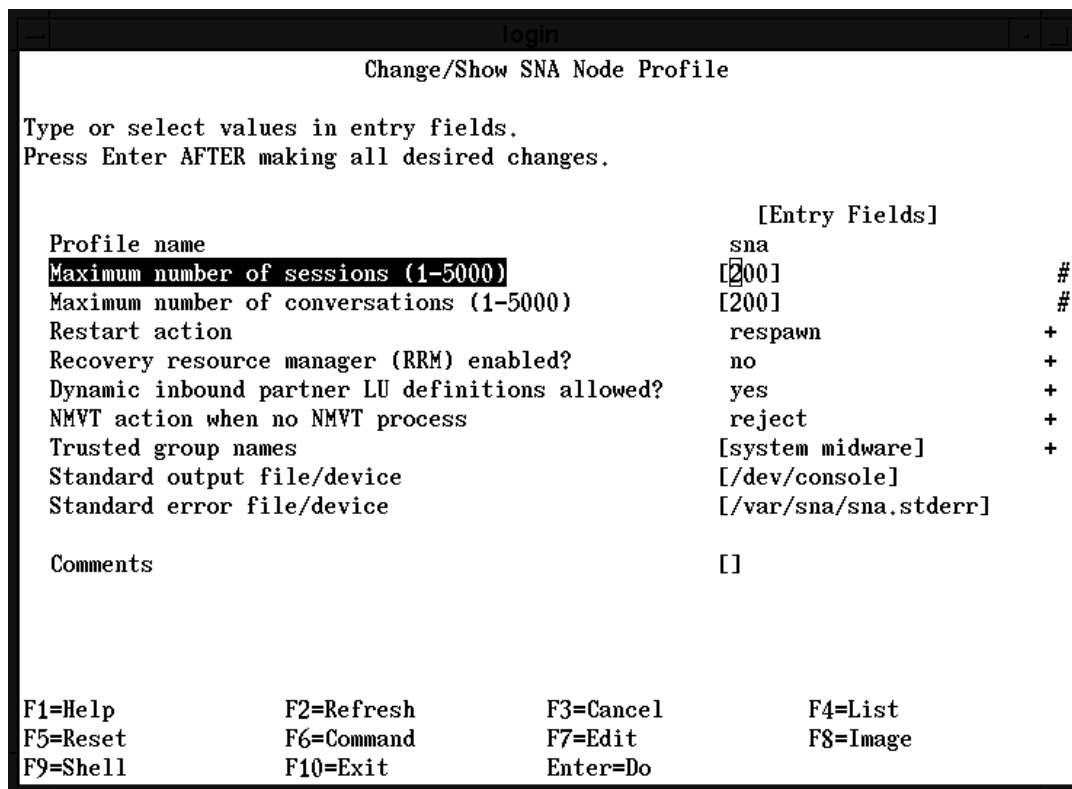


Figure 17. SNA System Defaults

3. Modify the sessions and conversations values if 200 is not enough. To calculate this number, multiply the maximum number of clients by the maximum number of data sources.
Keep in mind that these are global limits and all LUs are included. Also, each LU type 1, 2, and 3 defined counts as two sessions and two conversations.
4. Press F3 to return to the **Advanced Configuration** menu.

Verify Configuration Profiles

Next, you must verify the configuration profiles. From the **Advanced Configuration** menu:

1. Select **Verify Configuration Profiles**.
2. Change the update action default from none to `dynamic_update`.
3. Leave the remaining fields unchanged.

4. Press Enter and check the output screen for syntax errors. The profiles should verify with no errors.
5. Press F3 to return to the **Advanced Configuration** menu.



When you have finished setting up APPC network communications, continue the configuration process. If you are configuring for access within the heterogeneous environment, go to “Step 2: Update the Node Directory”. If you are configuring for access outside the heterogeneous environment, go to “Step 2: Update the Node Directory” on page 96.

Step 2: Update the Node Directory

Update the node directory by using either the command line processor (CLP) or the Database Director utility. To catalog a node, you must have SYSADM authority.

The following example shows the proper syntax for updating the node directory from the CLP. Use the first command for APPC configurations; use the second command for TCP/IP configurations. For example:

```
CATALOG APPC NODE db2node REMOTE db2cpidj SECURITY program
```

or

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

where:

NODE is an arbitrary alias for the node being cataloged.

REMOTE

For APPC configurations corresponds to the LU 6.2 Side Information Profile being used to communicate with the DRDA data source. 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 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...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 92 . This value is case sensitive.
- Set DATABASE to the name of a database at the DB2 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 “drda” if using APPC, or to “drdaIP” if using TCP/IP. This value is case sensitive and needs to be enclosed in delimiters.
- Set AUTHID to an authorization name at the data source that has BINDADD 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 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 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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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, 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.sa.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.sa 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.

Step 4: Test the Connections to the Data Source

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:

1. Connect to the DataJoiner database.
2. 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.

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)
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 “Configuring SNA Server/6000 for DRDA Data Sources” on page 78 or “Configuring TCP/IP for DRDA3 Data Sources” on page 76 for instructions.

Step 2: Update the Node Directory

The node directory contains entries for some types of remote servers that your workstation can access. Update the node directory by using either the command line processor (CLP) or the Database Director utility.

The following example shows the proper syntax for updating the node directory by using the CLP. Use the first command for APPC configurations; use the second command for TCP/IP configurations:

```
CATALOG APPC NODE db2node REMOTE db2cpidj SECURITY program
```

or

```
CATALOG TCPIP NODE db2node REMOTE tcphost SERVER db2prod
```

where:

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

REMOTE

For APPC configurations corresponds to the LU 6.2 Side Information Profile being used to communicate with the DRDA data source. 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 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. SERVER is used only for TCP/IP configurations.

For more information about the **CATALOG...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 DB2/MVS or DB2 for OS/390 data sources, it is the LOCATION in the Bootstrap data set (BSDS) of the subsystem being accessed.

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 server
```

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 96 .

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

Step 5: Bind Packages used by DataJoiner and Other Applications

To simplify the bind procedure, bind files are grouped together in different .lst files, each one being specific to a 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 for AIX 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 list:

DRDA Server	<DRDA_server>.lst
MVS	ddcsmvs.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:

```
db2 connect to db2mvsa
db2 bind HOME/sql1lib/bnd/@ddcsmvs.lst blocking all grant public
db2 bind HOME/sql1lib/bnd/@applycs.lst isolation cs blocking all grant public
db2 bind HOME/sql1lib/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_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 /sql1lib/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 might 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 *DB2 Installing and Using AIX Clients* and the *IBM*
| *DDCS User's Guide* (SC09-1923-00).

Chapter 8. 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.

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 and user information
4. Test connections

Step 1: Configure Network Communications

DataJoiner supports TCP/IP, IPX/SPX, and APPC. Configuring DataJoiner to any of these is similar to configuring a client to DataJoiner. Configuration instructions for these protocols are provided in the following sections:

- TCP/IP; see "Configuring TCP/IP for DB2RA Data Sources"
- IPX/SPX; see "Configuring IPX/SPX for DB2RA Data Sources" on page 104
- APPC; see "Configuring APPC for DB2RA Data Sources" on page 107

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 or DB2 Universal Database data source. This section can be used as a guide for configuring any DB2RA data source that is accessed through TCP/IP. The instructions here are similar to those in "Configuring CAE 2.1.1 for AIX or SDK 2.1.1 for an AIX TCP/IP Client" on page 183. 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.

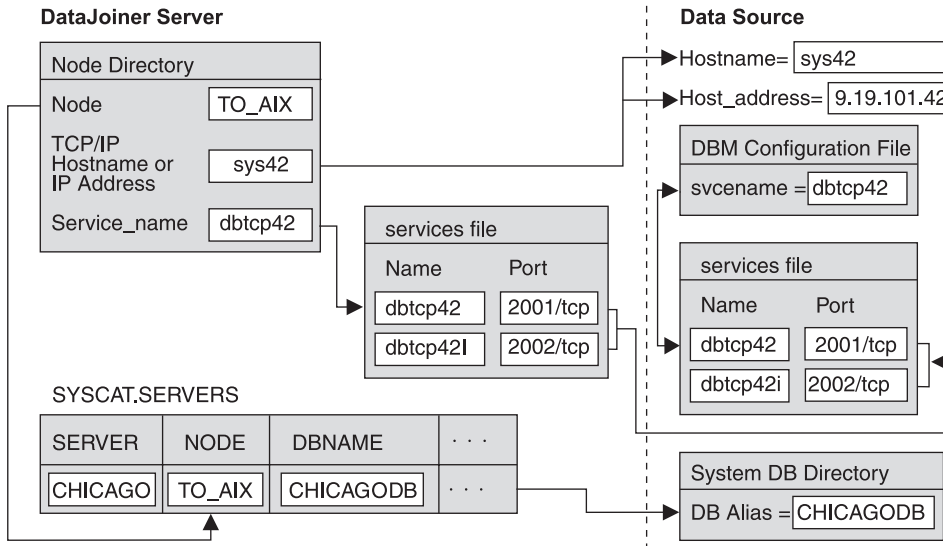


Figure 18. DB2RA Data Source through TCP/IP

The values that are used in this section are in Figure 18. We also provide a blank template in Figure 19 on page 104. 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 configure 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 `/etc/services` file on the data source's system. In the example, the entries would look like:

```
dbtcp42      3700/tcp      # Data source primary connect port
dbtcp42i    3701/tcp      # Data source interrupt port
```

Note: 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 `/etc/services` with the port numbers. All DB2RA data sources communicate with TCP/IP clients through ports that are defined in `/etc/services`. 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 `/etc/services` file. In the example, the entries would be:

```
dbtcp42      3700/tcp      # Data source primary connect port
dbtcp42i    3701/tcp      # Data source interrupt port
```

Alternately, you can specify the port numbers on the server parameter of the **CATALOG TCPIP NODE** command.

2. Synchronize the `/etc/services` file and the `inetd` daemon by issuing the following AIX commands:

```
inetimp
refresh -s inetd
```

This step is not required for AIX V4 or higher.

3. As root user, update `/etc/hosts` for the data source workstation, if necessary. Whether you must update `/etc/hosts` 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 `/etc/hosts`. Otherwise, you need an entry in `/etc/hosts` for the remote host name. See your network administrator to learn how your network is configured.

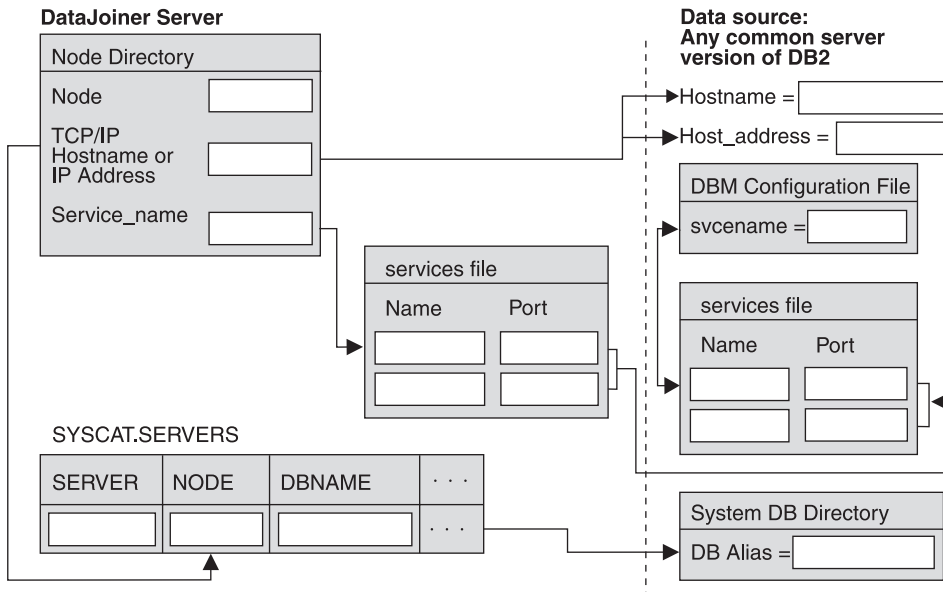


Figure 19. DB2RA Data Source Planning Template for TCP/IP



When you have finished setting up TCP/IP network communications, continue the configuration process. If you are configuring for access within the heterogeneous environment, go to “Step 2: Update the Node Directory” on page 125. If you are configuring for access outside the heterogeneous environment, go to “Step 2: Update the Node Directory” on page 129.

Configuring IPX/SPX for DB2RA Data Sources

DataJoiner provides IPX/SPX communication support through the IPX/SPX support option (db2_02_01.cs.ipx).

To enable IPX/SPX support for a 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 DataJoiner

IPX/SPX support is provided by the following products:

- IBM Netware for AIX Version 3.11B (this is the complete product which includes a file server)
- IBM AIX NetBIOS and IPX Support/6000

NetWare 4.x Directory Services is not supported in this release of DataJoiner. You should use the bindery emulation feature of NetWare 4.x.

You can access DataJoiner through IPX/SPX as follows:

- File Server Addressing

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

- Direct Addressing

The client connects to DataJoiner by directly specifying the NetWare internetwork address of the server (bypassing the NetWare file server). 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 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 DataJoiner 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 instance on a machine. It must also be unique among all IPX/SPX applications running on the DataJoiner machine to ensure that DataJoiner is able to listen to incoming IPX/SPX connections 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 that use either of the following addressing schemes:

- File server addressing and direct addressing
- File server addressing only

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

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

- 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 DataJoiner will be accessed by clients using only direct addressing.

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

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

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

Registering the DB2 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. To register the DataJoiner server, issue the following command:

```
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.

Notes:

1. This command must be issued locally from a DataJoiner server (it is not supported remotely) *once* before the first **db2start** for each DataJoiner instance.
2. If you move DataJoiner 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.



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 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).

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.



When you have finished setting up IPX/SPX network communications, continue the configuration process. If you are configuring for access within the heterogeneous environment, go to "Step 2: Update the Node Directory" on page 125 . If you are configuring for access outside the heterogeneous environment, go to "Step 2: Update the Node Directory" on page 129.

Configuring APPC for DB2RA Data Sources

This section shows how to use APPC to establish a connection from DataJoiner to a DB2 for OS/2 data source. Figure 20 on page 108 gives an overview of the configuration values that are used on the DataJoiner and OS/2 workstations in this example. Figure 21 on page 109 is a blank planning template you can use to plan your environment.

DataJoiner

DB2 for OS/2 V2 Data Source

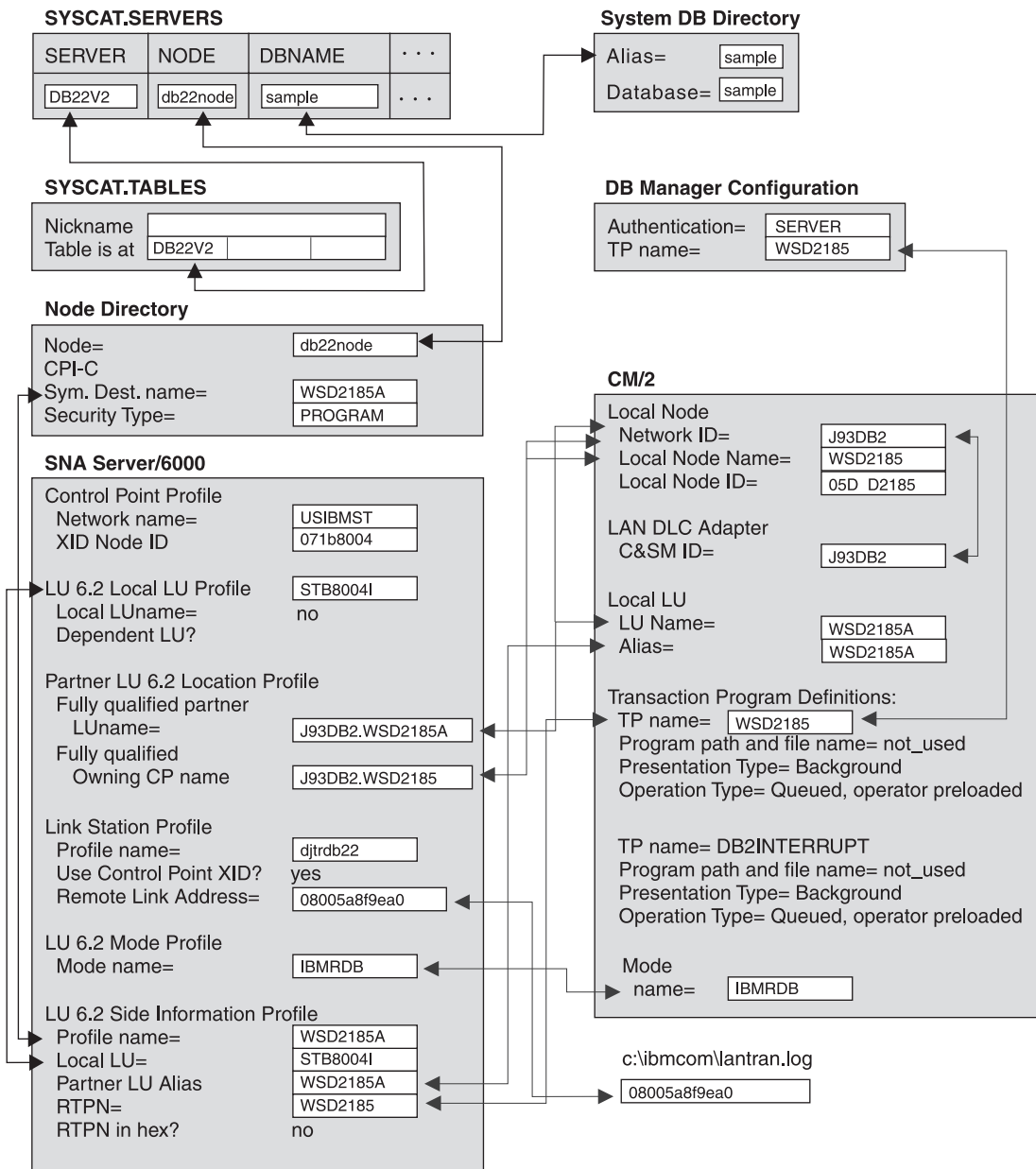


Figure 20. APPC Sample Values: Connecting DataJoiner to a DB2 for OS/2 V2 Data Source

DataJoiner

SYSIBM.SYSSERVERS

SERVER	NODE	DBNAME	...
<input type="text"/>	<input type="text"/>	<input type="text"/>	...

SYSIBM.SYSTABLES

Nickname
Table is at

Node Directory

Node=
CPI-C
Sym. Dest. name=
Security Type=

SNA Server/6000

Control Point Profile
Network name=
XID Node ID

LU 6.2 Local LU Profile
Local LUname= no
Dependent LU?

Partner LU 6.2 Location Profile
Fully qualified partner LName=
Fully qualified Owing CP name

Link Station Profile
Profile name=
Use Control Point XID? yes
Remote Link Address=

LU 6.2 Mode Profile
Mode name=

LU 6.2 Side Information Profile
Profile name=
Local LU=
Partner LU Alias
RTPN=
RTPN in hex? no

DB2 for OS/2 V2 Data Source

System DB Directory

Alias=
Database=

DB Manager Configuration

Authentication=
TP name=

CM/2

Local Node
Network ID=
Local Node Name=
Local Node ID=

LAN DLC Adapter
C&SM ID=

Local LU
LU Name=
Alias=

Transaction Program Definitions:
TP name=
Program path and file name= not_used
Presentation Type= Background
Operation Type= Queued, operator preloaded

TP name= DB2INTERRUPT
Program path and file name= not_used
Presentation Type= Background
Operation Type= Queued, operator preloaded

Mode
name=

c:\ibmcom\lantran.log

Figure 21. APPC Planning Template: Connecting DataJoiner to a DB2 for OS/2 V2 Data Source

On the Data Source Workstation

To configure the OS/2 data source machine:

1. Check the data source database authentication value.
2. Configure Communications Manager/2.
3. Update the database manager configuration file for APPC.
4. Set DB2COMM for APPC.

Details for each step follow.

Step 1, Check the Data Source Database Authentication Value: In this example, authentication is set to SERVER. This value is contained in the database manager configuration file. It must be consistent with the values for security in the NODE directory and PASSWORD column in SYSCAT.SERVER_OPTIONS on the DataJoiner workstation. Refer to the *DataJoiner Administration Guide* for details.

Step 2, Configure Communications Manager/2: To use the APPC protocol, you need to create or customize a number of profiles at the data source. If you have a token ring LAN, you must set up the following profiles:

- Local Node Characteristics Profile
- LAN DLC Profile
- Transaction Program Definition Profiles
- Mode Definition Profile
- Local LU Profile

To configure Communications Manager/2, follow these steps. These steps assume you are using Communications Manager/2 Version 1.11.

1. Double-click the **Communications Manager/2** icon.
2. Double-click the **Communications Manager Setup** icon.
3. Click **Setup**. The Open Configuration panel appears.
4. Specify a configuration name. You can enter a new configuration name or select an existing configuration file. Click **OK**.
5. In the Communications Manager Configuration Definition window, select the **Options** menu, then select **Configure any profile or feature** from the drop down list.

The following sections explain how to set up each profile.

Local Node Characteristics: The local node characteristics profile contains the parameters that are needed to identify the workstation to the SNA network. To set up the local node characteristics profile:

1. Select **SNA local node characteristics**.
2. Click **Configure**.
3. Enter the following values:

- Network ID: J93DB2
 - Local node name: WSD2185
This is the name other workstations know this OS/2 workstation by. It must be unique in the network.
 - Node type: End node - no network node server
 - Local node ID (hex): 05D D2185
If the workstation is connecting to a host through a 37x5 controller, you must enter this value (it is the IDBLK and IDNUM values concatenated). Otherwise, it is optional.
4. Continue to the Local Node Options dialog box. Click **Options**. Use the following values on the Local Node Options:
 - Local node alias name: WSD2185
 - Maximum compression level: None
 - Maximum compression tokens: 0
 - Activate Attachment Manager at Start up: selected
 5. Click **OK** to return to the **Communications Manager Configuration List**.

LAN DLC Profile: The LAN DLC Adapter profile identifies the type of LAN adapter and the adapter characteristics.

To set up the LAN DLC profile:

1. Select the **DLC - Token-ring or other LAN types**.
2. Click **Configure**.
3. Enter the C&SM LAN ID value:
J93DB2

This value matches the network ID specified on the SNA local node characteristics profile.

4. Click **OK**.

Local LU: These steps define a local LU rather than using the Local Node name for this connection.

1. Select the **SNA features**.
2. Click **Configure**.
3. Select the **Local LUs**.
4. Click **Create**.
5. Enter the following values:
 - LU Name: WSD2185A (must be unique to this workstation)
If this workstation is communicating with a VTAM host, you can share the local LU. In that case, the value comes from the LU definitions of the local VTAM host as shown in Figure 13 on page 84.

- Alias: WDS2185A (referenced in the SNA Server/6000 configuration as the partner LU alias).

Independent LU is turned on. Use this local LU as your default LU is checked.

6. Click **OK** to return to **SNA features** window.

Transaction Program Definitions: The transaction program (TP) definition provides the information necessary to start a TP on the local node, and specifies the properties of the TP. You must define two definitions for DataJoiner to access DB2 for OS/2 V2 as a data source.

Because the TP is operator-preloaded, the OS/2 program path and file name field is not used. However you must enter something when you configure the TP, because the field is a mandatory input field. To define TP parameters:

1. Select the **Transaction program definitions** from the features list.
2. Click **Create**.
3. Enter the appropriate values for the TPs that you are configuring.
4. Click **Continue**. See “Additional TP Parameters” to complete this step.

TP for CAE V2 Database Connections and Interrupts: Use the following TP definition to handle database connections from CAE V1 clients, such as DataJoiner:

- Transaction program (TP) name: WSD2185
- OS/2 program path and file name: not_used

To configure TP for CAE V2:

1. Enter a value for the TP name that matches the value that is specified on the DataJoiner workstation in CPI-C Side Information.
2. Click **Continue**.
3. Configure the TP definition with the following TP parameters:
 - Presentation type: Background.
 - Operation type: Queued, operator preloaded.
4. Click **OK**.

TP for CAE V1 Interrupts: The following TP definition is used to handle database interrupts from CAE V1 clients, such as DataJoiner. Use the steps above to create this TPN with the following values:

- Transaction program (TP) name: DB2INTERRUPT
- OS/2 program path and file name: not_used

After entering the value, click **Continue**.

Additional TP Parameters: Configure the TP definitions with the following values:

1. Presentation type: Background.
2. Operation type: Queued, operator preloaded.

3. Click **OK**.

Mode Definition: You can use an IBM-defined mode, or add a new mode definition that is appropriate for your application. The mode name, IBMRDB in the example, must also exist on the DataJoiner workstation in SNA Services/6000. To enter modes:

1. Select **Modes**.
2. Click **Create**.
3. Enter the values:
 - Mode name: IBMRDB
 - Class of service: #CONNECT
 - Mode session limit: 30
 - Minimum contention winners: 15
 - Receive pacing window: 4
 - Maximum RU size: 4096
4. Click **OK**.

These values are similar to the values that are shown for the mode profile in Figure 27 on page 121. Refer to Communications Manager/2 online help for detailed information.

APPC Security Considerations: Communications Manager/2 should be configured appropriately to accept the type of APPC security requested by the client. The APPC security type for the data source can be one of the following:

- NONE
- PROGRAM
- SAME

For details, refer to *DB2 for OS/2 Installation and Operations Guide (S20H-4785)*. The example assumes a security type of PROGRAM. Nothing more needs to be done.

Step 3, Update the Database Manager Configuration File for APPC: Specify the transaction program name as the value for the *tpname* parameter in the database manager configuration file. *Tpname* must be the same as the transaction program name that is configured in the transaction program profile that handles database connections from DB2 Client Application Enabler Version 1 clients.

You can update the configuration file by using either:

- The Command Line Processor. For example, you can issue the following command:
`db2 update database manager configuration using tpname`

where *tpname* is WSD2185 in the example.

- The DB2 Database Director

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

Step 4, Set DB2COMM for APPC: You must set a value of APPC in the DB2COMM environment variable in the config.sys file to enable APPC support when a **db2start** command is issued. For example:

```
SET DB2COMM=APPC
```

See the *Database/2 for OS/2 Installation and Operation Guide* (S20H-4786) for more information.

On the DataJoiner Server Workstation

To configure the DataJoiner server workstation, you must configure SNA Server/6000.

Configure SNA Server/6000: Defining SNA resources requires profile configuration. Smitty provides the easiest way to configure profiles. This section gives one method of configuring the profiles. Refer to the *AIX SNA Server/6000 Configuration Reference, SC31-7014*, for detailed information. To access a DB2RA data source, you must configure the following:

- Physical Connections
- Links
- Control Point Profile
- LU 6.2 Sessions

The steps that follow assume you are using SNA Server/6000 V2:

1. To access the SNA profiles with smitty, enter `smi tty` at the DataJoiner server workstation.
2. Select **Communications Applications and Services**.
3. Select **SNA Server/6000**.
4. Select **Configure SNA Profiles**.
5. Select **Advanced Configuration**. From **Advanced Configuration**, you can define links, sessions, control points, and profiles.

Physical Connections: SNA Server/6000 supports five communication adapters. The examples in this section are for configuring a token-ring adapter.

Links: Links depend on the type of physical connection to your system. This example is for a token-ring adapter. SNA Server/6000 V2 supports multiple call and listen link stations on one token-ring adapter. For this reason you must ensure that you have one of each of the following profiles:

- Token-Ring SNA DLC Profile (usually one per workstation)
- Token-Ring Link Station Profile (one per link to each data source host)

The steps that follow assume that these profiles must be created:

1. Invoke `smit SNA` on the DataJoiner workstation.
2. From the **Advanced Configuration** window, select **Links**.
3. Select **Token-Ring**.

4. Select **Token-Ring SNA DLC**.
 5. Select **Add a Profile**.
- The **Change/Show Token Ring SNA DLC Profile** window opens.

```

login
Change/Show Token Ring SNA DLC Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
Current profile name                       tok0.00001
New profile name                           [ ]
Data link device name                       [tok0]
Force disconnect time-out (1-600 seconds)  [120]
User-defined maximum I-Field size?        no
If yes, Max. I-Field size (265-30729)     [30729]
Max. num of active link stations (1-255)   [100]
Number reserved for inbound activation     [0]
Number reserved for outbound activation    [0]
Transmit window count (1-127)             [16]
Dynamic window increment (1-127)          [1]
Retransmit count (1-30)                   [8]
Receive window count (1-127)              [8]
Ring access priority                       0

[MORE...29]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command      F7=Edit       F8=Image
F9=Shell     F10=Exit        Enter=Do

```

Figure 22. Token-Ring SNA DLC Profile

When filling out the **Add Token-Ring SNA DLC Profile** menu, consider the following:

1. **Data link device name** is the RISC's token-ring card that is associated with this link. The default is tok0. If the RISC System/6000 has only one Token-Ring adapter, this field is tok0.
2. The default **Current profile name** is tok0.00001 for token-ring adapter card 0.
3. This configuration assumes that **Dynamic link stations supported?** is set to the default value of yes. Note that you must scroll to see this field.
4. The remaining fields can use the default values provided by SNA Server/6000.
5. Press Enter after filling out the profile and check the output screen for syntax errors.
6. Press the F3 key to back up the Token-Ring physical unit menu.

Now you must select **Add a Profile** from the **Token-Ring Link Station** menu choice.

```

login
Change/Show Token Ring Link Station Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...1]
New profile name [ ] [Entry Fields]
Use Control Point's XID node ID? yes +
  If no, XID node ID [*]
* SNA DLC Profile name [tok0.00001] +
Stop link station on inactivity? no +
  If yes, Inactivity time-out (0-10 minutes) [0] #
LU address registration? no +
  If yes,
  If yes, LU Address Registration Profile name [ ] +
Trace link? no +
  If yes, Trace size long +

Adjacent Node Address Parameters
Access routing link_address +
[MORE...39]

F1=Help      F2=Refresh   F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit      F8=Image
F9=Shell     F10=Exit     Enter=Do
  
```

Figure 23. Link Station Profile Screen

At the link station profile screen:

- Provide profile names for the link station profile.
- Create a descriptive name. Use a name that describes the data source host for which this profile is being added.
- The XID node ID can be coded in the Control Point Profile or the Link Station Profile. In the example, it is coded in the link station The default value for **Use Control Point's XID node ID?** is yes.
- Enter the SNA DLC Profile name that you previously used (that is, tok0.00001).
- Set the value of the Adjacent Node Address Parameter to "link_address."
- The **Remote link address** field is the universal or locally administered token-ring address of the destination host. For OS/2, the value is located in C:\IBMCOM\LANTRAN.LOG. You must scroll to see this field.
- Remaining fields use default values provided by SNA Server/6000.
- Press Enter after filling out the profile and check the output screen for syntax errors.

- Press F3 to return to the **Advanced Configuration** menu.

Control Point Profile: To configure a control point profile:

1. From the **Advanced Configuration** menu, select **Control Point**.
2. Select **Change/Show a Profile**.

In most cases, the profile is already created for 3270 LU Type 2 connections and can be used again. If the profile does not exist, choose **Add a Profile**. The Change/Show a Profile window opens:

```

login
Change/Show Control Point Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Profile name                    node_cp
  XID node ID                      [071b8004]
  Network name                     [USIBMST]
  Control Point (CP) name          [DJCP]
  Control Point alias              [DJCP]
  Control Point type               appn_end_node      +
  Maximum number of cached routing trees [500]      #
  Maximum number of nodes in the TRS database [500]      #
  Route addition resistance         [128]          #

  Comments                        []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit        Enter=Do

```

Figure 24. Control Point Profile

In the Change/Show a Profile window:

3. The default **Profile name** is node_cp.
4. The **XID node ID** can be coded in the Control Point Profile or the Link Station Profile.
5. The **XID node ID** field is used during an exchange ID command sequence to identify the RISC System/6000 to VTAM. VTAM uses the XID value to match against VTAM's Switched Major Node's value and assign a physical unit (PU) definition to the RISC System/6000.

The matching occurs on VTAM's PU statement by combining the IDBLK and the IDNUM fields and comparing to the XID value sent by the RISC System/6000.

6. Set the **XID node ID** field equal to the IDBLK and IDNUM fields. The switched node definitions are managed by a VTAM or network administrator. Only one system can use a PU definition at a time, so each DataJoiner server needs a unique PU definition with unique IDBLK and IDNUM values.
7. The **NETWORK name** should be equal to the value of the owning VTAM's Startup List NETID parameter. This is the local VTAM host which has the PU and LU definitions.
8. Assign a name to the **Control Point name** and **Control Point alias** fields. This value is used in an APPN network. It should be set to the value of the CPNAME parameter that is associated with the PU in the VTAM Switched Major Node Definition, if one exists.
9. The **Control Point type** can be appn_end_node (EN) or appn_network_node (NN). Type appn_end_node unless in an APPN network. If in an APPN network, type is appn_network_node (NN).
10. Use default values provided by SNA Server/6000 in the remaining fields.
11. Press Enter after filling out the profile and check the output screen for syntax errors.
12. Press F3 to return to the **Advanced Configuration** menu.

LU 6.2 Sessions: LU 6.2 sessions require the creation of the following profiles:

- Local LU
- Partner LU 6.2 Location Profile
- Mode
- LU 6.2 Side Information

Local LU: After physical unit (PU) definitions are complete, you must define LU 6.2 definitions or use an existing LU 6.2. To define LU 6.2 definitions:

1. From the **Advanced Configuration** menu, select **Sessions**.
2. Select **LU 6.2**.
3. Select **LU 6.2 Local LU**.
4. Select **Change/Show a profile..**

The **Change/Show LU 6.2 Local LU Profile** window opens.

```

login
Change/Show LU 6.2 Local LU Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Current profile name          [Entry Fields]
                               ludj1
New profile name             []
Local LU name                 [STB8004I]
Local LU alias                [STB8004I]
Local LU is dependent?       no
                               +
  If yes,
    Local LU address (1-255)   []
                               #
    System services control point
      (SSCP) ID (*, 0-65535)   [*]
    Link Station Profile name  []
                               +
Conversation Security Access List Profile name  []

Comments                     []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command      F7=Edit       F8=Image
F9=Shell     F10=Exit        Enter=Do

```

Figure 25. LU 6.2 Local LU Profile

- There are many options for a local LU. This example uses one that already exists on the workstation, as shown in Figure 13 on page 84.
- Be sure the **Local LU is dependent?** field is set to No. No is the default.
- Leave remaining fields alone.
- Press Enter after filling out the profile and check the output screen for syntax errors.
- Press F3 to return to the LU 6.2 menu.

Partner LU 6.2 Location Profile: Next, configure the Partner LU 6.2 location from the **LU 6.2** menu:

1. Select **Partner LU 6.2 Location** (not LU 6.2 Partner LU).
2. Choose **Add a Profile**.

Change/Show Partner LU 6.2 Location Profile			
Type or select values in entry fields. Press Enter AFTER making all desired changes.			
		[Entry Fields]	
Current profile name		WSD2185A	
New profile name		[]	
Fully qualified partner LU name		[J93DB2.WSD2185A]	
Partner LU location method		owning_cp	+
If owning_cp,			
Fully qualified owning Control Point (CP) name		[J93DB2.WSD2185]	
Local node is network server for LEN node?		no	+
Fully qualified network node server name		[]	
If link_station,			
Local LU name		[]	
Link Station Profile name		[]	+
Comments		[]	
F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 26. Partner LU 6.2 Location Profile

- The Partner LU 6.2 Location Profile points to the partner LU and the control point where the partner is located, and defines the method of locating the partner.
- For the profile name, use a descriptive name. Consider using a name that defines the remote host, such as the partner LU name.
- The **Partner LU name** is the LU that defines the local LU on the remote host; an OS/2 workstation in this example.
- The **Partner LU location method** that is selected with this example is `owning_cp`.
- The control point name is the network ID and local node name in Communications Manager on the OS/2 workstation.
- Accept default values for the remaining fields.
- Press Enter after filling out the profile and check the output screen for syntax errors.
- Press F3 to return to the **LU 6.2** menu.

Mode: Modes, such as `IBMRDB`, must be unique, but they can be shared by multiple applications. The same mode can be used for remote DataJoiner clients and remote data sources. In this example, assume that the mode already exists. To access it:

1. Select **LU 6.2 Mode, Change/Show a Profile**. Mode names, such as IBMRDB, must be unique, but they can be used by multiple locations. Select **Change/Show a Profile** to view an existing profile.
2. Press F4 to see a list of all existing mode profiles. Highlight IBMRDB. Press enter.

```

login
Change/Show LU 6.2 Mode Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Current profile name          [Entry Fields]
                               IBMRDB
New profile name              []
Mode name                     [IBMRDB]
Maximum number of sessions (1-5000) [32]
Minimum contention winners (0-5000) [16]
Minimum contention losers (0-5000) [16]
Auto activate limit (0-500)      [0]
Upper bound for adaptive receive pacing window [16]
Receive pacing window (0-63)     [3]
Maximum RU size (128,...,32768: multiples of 32) [2048]
Minimum RU size (128,...,32768: multiples of 32) [256]
Class of Service (COS) name      [#CONNECT]

Comments                      []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 27. Mode Profile

Keep the following in mind:

- The **Mode name** field must match the mode name that is specified in the Side Information Profile, which is defined in a later step.
The **Mode name** field is the mode name that is to be sent during the LU-to-LU bind. The remote OS/2 workstation must have the same mode name defined. Enter the mode name in upper case.
The profile name and the MODE name do not need to be the same.
- The mode profile defines the maximum number of concurrent sessions that are allowed between DataJoiner and an application server.

The **Maximum number of sessions** field determines how many concurrent client connections are to be allowed to a particular application server. If DataJoiner has remote clients, ensure that the minimum number of contention winners is greater than 0. Together, the winners and losers cannot exceed the maximum number of sessions.

- The pacing fields set how many LU frames are to be sent or received before an acknowledgment is required.
- The RU size sets the largest and smallest Request/Response unit size that can be handled by an application. The suggested maximum RU size is 4096 bytes. The suggested minimum RU size is 1024 bytes. The maximum RU size allowed for SNA Server/6000 is 32,768 bytes. The minimum RU size allowed for SNA Server/6000 is 128. The RU Size must be specified in multiples of 32 bytes.
- The suggested Class of Service Name is #CONNECT. It is the default.
- Press Enter after filling out the profile and check the output screen for syntax errors.
- Press F3 to return to the **LU 6.2** screen.

LU 6.2 Side Information: The LU 6.2 Side Information Profile is pointed to by the Node directory. The profile points to the partner LU (DRDA AS). You must create a Side Information Profile for each DRDA Application Server subsystem. From the **LU 6.2** screen:

1. Select **LU 6.2 Side Information**.
2. Select **Add a Profile**.


```

Change/Show LU 6.2 Side Information Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Current profile name           [Entry Fields]
                               WSD2185A
New profile name               []
Local LU or Control Point alias [STB8004I]      +
Provide only one of the following:
  Partner LU alias             [WSD2185A]      +
  Fully qualified partner LU name []
Mode name                      [IBMRDB]      +
Remote transaction program name (RTPN) [WSD2185]
RTPN in hexadecimal?          no              +

Comments                       []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 28. LU 6.2 Side Information Profile

- Choose a descriptive profile name. It will be referenced later in “Step 2: Update the Node Directory” on page 125.
- The local LU comes from the LU 6.2 Local LU Profile. In the example, it is STB8004I.
- The fully qualified partner LU name consists of a Networkid.PartnerLUname which, in the example, is “USIBMST.ST11DB2A” and is the Partner LU Alias. It is the alias that is defined for the Local LU on the data source.
- The “Mode name” is the name of the mode that is passed during LU bind processing. The name must equal a “Mode name” in a LU 6.2 Mode profile and must be entered in uppercase.
- The “Remote Transaction Program name” field is the name of the Application Server transaction program. For DB2 for OS/2 V2, this is the TPname on the OS/2 workstation that is defined to support database connections from CAE V1 clients. The example used WSD2185.
- Set the “RPTN in a hexadecimal?” field to “no.”
- Press Enter after filling out the profile and check the output for syntax errors.
- Press F3 to return to the **Advanced Configuration** menu.

SNA System Defaults: The SNA Node Profile is used to set global limits on the number of concurrent sessions and conversations. This profile might already exist. From the **Advanced Configuration** menu:

1. Select **SNA System Defaults**.
2. Select **Change/Show a Profile**.

```

login
Change/Show SNA Node Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Profile name                      sna
Maximum number of sessions (1-5000) [200] #
Maximum number of conversations (1-5000) [200] #
Restart action                      respawn +
Recovery resource manager (RRM) enabled? no +
Dynamic inbound partner LU definitions allowed? yes +
NMVT action when no NMVT process    reject +
Trusted group names                 [system middleware] +
Standard output file/device          [/dev/console]
Standard error file/device           [/var/sna/sna.stderr]

Comments                            []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Figure 29. SNA System Defaults

- Modify the sessions and conversations values if 200 is not enough. Multiply the maximum number of clients by the maximum number of data sources to calculate this number.
Keep in mind that these are global limits and all LUs are included. Also, each LU type 1, 2, and 3 defined counts as two sessions and two conversations.
- Press F3 to return to the **Advanced Configuration** menu.

Verify Configuration Profiles: Next, you must verify the configuration profiles. From the **Advanced Configuration** menu:

1. Select **Verify Configuration Profiles**.
2. The update action default is none. Change it to `dynamic_update`.
3. Leave the remaining fields unchanged.
4. Press enter and check the output screen for syntax errors. The profiles should verify with no errors.
5. Press F3 to return to the **Advanced Configuration** menu.



When you have finished setting up APPC network communications, continue the configuration process. If you are configuring for access within the heterogeneous environment, go to “Step 2: Update the Node Directory”. If you are configuring for access outside the heterogeneous environment, go to “Step 2: Update the Node Directory” on page 129.

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: 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 125. 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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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 db2v2cs1  
AUTHID SMITH PASSWORD JONES
```

where:

- benham is a local DataJoiner authorization name.
- db2v2cs1 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 db2v2cs1
SETTING 'Y'
```

where:

- pushdown is the option that controls whether or not DataJoiner will allow remote data sources to evaluate operations locally.
- db2v2cs1 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, 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 db2v2cs1.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.
- db2v2cs1 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.

Step 4: Test the Connections to the Data Source

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:

1. Connect to the DataJoiner database.
2. 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.

Accessing DB2RA Data Sources Outside the Heterogeneous Environment

You might want to configure DataJoiner to access DB2RA data sources outside the heterogeneous environment for two reasons:

- So applications such as Replication and Replication Administration can connect directly to DB2RA databases without using DataJoiner nicknames or pass-through sessions.
- As an alternative means of binding the packages used by DataJoiner to these data sources (DataJoiner also binds these packages when a CREATE SERVER MAPPING statement is issued for a DB2RA data source).

The steps 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:

- TCP/IP; see “Configuring TCP/IP for DB2RA Data Sources” on page 101
- IPX/SPX; see “Configuring IPX/SPX for DB2RA Data Sources” on page 104
- APPC; see “Configuring APPC for DB2RA Data Sources” on page 107

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 TCP/IP 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

DataJoiner requires certain packages to be bound at the data source.

To simplify the bind procedure, bind files are grouped together in different *.lst* files, each one being specific to a server. You must first connect to the database, then issue the **bind** command against the database using the appropriate *.lst* file. When binding against a DB2 for OS/2, DB2 for AIX, DB2 for HP-UX, or DB2 Solaris server, use the *db2ubind.lst* and the *db2cli.lst* files, which are contained in */usr/lpp/djx_02_01_01/bnd*. Issue the following commands:

```
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 want 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 all users.

The following example shows a typical command sequence for binding packages at a DB2/UDB database:

```
connect to db2udba
bind HOME/sqllib/bnd/@db2cli.lst blocking all grant public
bind HOME/sqllib/bnd/@db2ubind.lst blocking all grant public
bind HOME/sqllib/bnd/@applycs.lst isolation cs blocking all grant public
bind HOME/sqllib/bnd/@applyur.lst isolation ur blocking all grant public
```

Notes about binding:

- The **bind** command must be run 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 might take a few minutes to complete.

When the bind is finished, you can uncatlog the database.

Chapter 9. Accessing Informix Data Sources

For DataJoiner to access Informix databases, it needs a data access module built by link-editing DataJoiner libraries with Informix libraries. To make this job easier, DataJoiner provides a sample makefile, `djxlink.makefile`, that contains examples of link-editing DataJoiner with Informix.

Two types of Informix data access modules can be built with `djxlink.makefile`: `informix` and `informix7`. Use the `informix` data access module to access Informix databases that use Informix-Net Version 5. Use the `informix7` data access module to access Informix databases that use `informix` Version 7.

Restriction:

You may access both an Informix Version 5 and an Informix Version 7 database through the Informix-Net Version 5 data access module. However, accessing a Version 5 database through the Version 7 data access module is not supported.

The following two sections assume 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.

Using the Informix-Net Version 5 Data Access Module

Follow these steps if you are using Informix-Net Version 5 to access Informix databases.

Prerequisite step:

With Informix Version 5, either the `.netrc` file or the `/etc/hosts` and `/etc/hosts.equiv` files need to be set up before a client can successfully request a connection to a remote server.

These steps provide examples for using the `.netrc` file. If you want to use the `/etc/hosts` and `/etc/hosts.equiv` files, see the *Informix-Net and Informix-Star Installation and Configuration Guide*.

1. If you have not already done so, install Informix-Net Version 5 and the Informix `esql/c` libraries on the system where DataJoiner is installed.
2. Follow the directions in the *Informix-Net/Informix-Star Installation and Configuration Guide* for creating a `sqlhosts` file under the `$INFORMIXDIR/etc` directory. Figure 30 on page 132 shows the general structure of an `sqlhosts` file.

Informix sqlhosts file

DBSERVER NAME	Nettype	Hostname	Service Name
:	:	:	:
dbinst1	olsoctcp	mvpdb2	ifmxonl7
prod	olsoctcp	glacier	ifmxstar4
online2	olsoctcp	mvpdb0	ifmxstar2
online1	olsoctcp	mvpdb0	ifmxstar1

SYSCAT.SERVERS view

server	node	dbname	type	version	protocol	...
ifmx1	online1	testdb	INFORMIX	5.03	informix	

Figure 30. Informix V5 Data Access Module

3. If you have not already done so, build an Informix version 5 data access module by running `djxlink.sh`, as described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

For DataJoiner to send requests to Informix, it needs a data access module (a file) that has been built by link-editing DataJoiner libraries with Informix libraries. If you are accessing multiple Informix databases using the Informix-Net Version 5 data access module, each database that you access requires a data access module.

When you run `djxlink.sh`, if you receive a message that indicates that the Informix 5 data access module could not be built, you will need to edit the sample makefile named `/usr/lpp/djx_02_01_01/lib/djxlink.makefile` so that the makefile works with your particular setup. Editing and running `djxlink.makefile` are also described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

If `djxlink.sh` fails to build the Informix version 5 data access module, login as root and edit `djxlink.makefile`. Find the section in the file that is titled “Accessing Data Sources Through the Informix Data Access Module (Informix Version 5)”.

- a. In the makefile, respecify the `DJX_INFORMIX_LIBRARY_PATH` variable to be the directory specified in step 1 on page 131. This directory should be the location of the libraries for your Informix Version 5 driver.

- b. The name of the Informix 5 database access module that is built by the makefile is `informix`. If you do not want to change this name, skip to the next step.

If you want to change the name, you need to change two lines in the makefile:

- Change the target name on the dependency line. For example, if you want the data access module to be named `ifmx5`, change the dependency line from this:

```
informix: libinformix.a informixEP.o informix.exp informix.imp
```

To this:

```
ifmx5: libinformix.a informixEP.o informix.exp informix.imp
```

- Change the output name of the link-edit statement. For example, change this:

```
ld -o informix \
```

To this:

```
ld -o ifmx5 \
```

- c. As root, link-edit the `informix` data access module library by using the supplied makefile. To complete this task, you need the following:

- The `informix` data access module library that is provided with DataJoiner
- The `informix.exp` file that is provided with DataJoiner
- The Informix esql/c libraries that are provided with the Informix esql product

Run the makefile by entering:

```
cd /usr/lpp/djx_02_01_01/lib
make -f djxlink.makefile informix
```

If you changed the name of the data access module in the makefile, then use that name with the **make** command. For example:

```
make -f djxlink.makefile ifmx5
```

For AIX 3.2.5 users:

If you are running AIX 3.2.5, and you are unable to create the data access module because of errors from unresolved symbols (`mulh`, `divss` and `divus`), then add the following line to your makefile in the Informix 5 data access module section:

```
-bI:/usr/lib/FCM/lowsys.exp
```

Run the makefile again.

Whenever you rerun `djxlink.makefile` for the `informix` data access module, you must stop and restart all DataJoiner instances that use the `informix` data access module.

4. For each of the remaining Informix databases you wish to access through Informix-Net Version 5, copy the newly created data access module to a new file (where each file has a unique name). The unique file name is the value you will

specify in the PROTOCOL column of SYSCAT.SERVERS for that Informix database. To copy the data access module from the /usr/lpp/djx_02_01_01/lib directory, enter:

```
cp original-data-access-module-name new-data-access-module-name
```

Where *original-data-access-module-name* is the unique name from step 133, and *new-data-access-module-name* is the name of the new data access module. For example:

```
cp informix informixb
```

Do steps 5 through 10 on page 136 for each DataJoiner instance that will use the informix data access module.

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

```
export INFORMIXDIR="informix-home-directory"
export SQLRMDIR="relay-module-directory"
export SQLRM="relay-module-name"
```

where:

informix-home-directory

Is the directory in which you installed the e/sqlc library in step 1

relay-module-directory

Is the directory containing the relay module, typically
informix-home-directory/lib

relay-module-name

Is the name of the relay module to use for the Informix-Net communication protocol.

Restrictions:

- Be sure there are no spaces on either side of the equal sign (=) in these export statements.
- If DataJoiner and the Informix server are installed on the same machine, do not set the SQLRM environment variable.

See the *Informix-Net/Informix-Star Installation and Configuration Guide* for more information on these environment variables.

6. Use the CREATE SERVER MAPPING DDL statement to define each Informix database you expect to access via Informix-Net Version 5. Specify a different protocol for each database. When performing server mapping:

- Choose a unique server name.
- Set NODE to the appropriate dserver name from the Informix sqlhosts file. This value is case sensitive. For an example, see Figure 30 on page 132.

Note that you do not need to use the DataJoiner **CATALOG NODE** command for any data sources that are accessed through the Informix-Net Version 5 data access module.

- Set DATABASE to the name of an Informix database at the Informix server. This value is case sensitive.
- Set TYPE to INFORMIX.
- Set VERSION to the version of Informix that the Informix server is running. For example, 5.0 for Informix 5.03 (only one-digit numbers and two-digit numbers separated by a decimal point are supported)
- Set PROTOCOL to the name of a data access module you built in step 4 on page 133 . For example, "informix". (See Figure 30 on page 132). This value is case sensitive.

A sample entry for Informix Version 5 is:

```
CREATE SERVER MAPPING FROM infmxa TO NODE "infrmx1" DATABASE "test_db"
TYPE informix VERSION 5.0 PROTOCOL "informix"
```

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

7. Update the .netrc file of the DataJoiner instance with the host name, userid, and password that DataJoiner will use to access the dserver name you specified in SYSCAT.SERVERS; for example:

```
machine infmxa1 login joeuser password brody
```

where:

infmxa1

Is the dserver name that is specified in the Informix sqlhosts file

joeuser

Is the login-id that is to be sent to the Informix server

brody Is the password for the specified login-id

If .netrc already contains an entry for this host, you do not need to add another. DataJoiner uses the same .netrc entry to access all Informix databases on a given system.

8. If you want the Informix data access module to start dynamically when the DataJoiner instance is started, update the db2profile of the DataJoiner instance to include an entry for the DJXCOMM variable. If you do not set the DJXCOMM variable, you will need to manually start the Informix data access module.

The name of each unique Informix data access module you created should be specified in the DJXCOMM variable. For example, if you have a data access module that is named `informix` and another data access module that is named `ifmx5`, the entry would be:

```
export DJXCOMM=informix,ifmx5
```

See “Step 2: Set Environment Variables” on page 57 for more information.

9. Invoke `db2profile` through either of the following methods:
 - From the DataJoiner instance login, run `.profile`. For example:

```
. .profile
```
 - Invoke `db2profile` directly. For example:

```
. /home/djinst1/sql1lib/db2profile
```
10. Recycle (**db2stop** followed by **db2start**) the DataJoiner instance.
11. 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 infmxa  
SETTING 'I'
```

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

12. 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.

13. 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 11 on page 136. 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.

Using the Informix Version 7 Data Access Module

Notes:

- AIX/6000 requires that the Portable Streams Environment be loaded in order to access Informix databases through Informix Version 7
- DataJoiner supports Informix Versions 5, 6, and 7.1. DataJoiner also supports Informix ALS Versions 5 and 6.

To configure DataJoiner access to Informix databases through Informix Version 7:

1. If you have not already done so, install the Informix esql/rt or equivalent libraries on the DataJoiner system.
2. Follow the directions in the *Informix—OnLine Administrator's Guide* to create an sqlhosts file under the \$INFORMIXDIR/etc directory. Figure 31 shows the general structure of an sqlhosts file.

Informix sqlhosts file

DBSERVER NAME	Nettype	Hostname	Service Name
:	:	:	:
dbinst1	olsoctcp	mvpdb2	ifmxonl7
prod	olsoctcp	glacier	ifmxstar4
online2	olsoctcp	mvpdb0	ifmxstar2
online1	olsoctcp	mvpdb0	ifmxstar1

SYSCAT.SERVERS view

server	node	dbname	type	version	protocol	...
infmtx1	online1	testdb	INFORMIX	7.2	informix	

Figure 31. Informix V7 Data Access Module

- |
- |
- |
3. If you have not already done so, build an Informix version 7 data access module by running `djxlink.sh`, as described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

For DataJoiner to send requests to Informix, it needs a data access module (a file) that has been built by link-editing DataJoiner libraries with Informix libraries. For Informix Version 7 databases, you need only one data access module regardless of the number of databases you will be accessing.

|

|

|

|

When you run `djxlink.sh`, if you receive a message that indicates that the Informix 7 data access module could not be built, you will need to edit the sample makefile named `/usr/lpp/djx_02_01_01/lib/djxlink.makefile` so that the makefile works with your particular setup. Editing and running `djxlink.makefile` are also described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

If `djxlink.sh` fails to build the Informix version 7 data access module, login as root and edit `djxlink.makefile`. Find the section in the file that is titled ‘Accessing Data Sources Through the Informix Data Access Module (Informix Version 7)’.

- a. In the makefile, respecify the `DJX_INFORMIX7_LIBRARY_PATH` variable to be the directory specified in step 1 on page 137. This directory should be the location of the libraries for your Informix Version 7 driver.

Attention:

If you are using the ESQL/C library, see the `informix7c` sample in the `djxlink.makefile`.

The unique file name is the value you specify as the `PROTOCOL` value in the `CREATE SERVER MAPPING DDL` statement.

- b. The name of the Informix 7 database access module that is built by the makefile is `informix7`. If you do not want to change this name, skip to the next step.

If you want to change the name, you need to change two lines in the makefile:

- Change the target name on the dependency line. For example, if you want the data access module to be named `ifmx7`, change the dependency line from this:

```
informix7: libinformix7.a informixEP.o informix.exp informix.imp
```

To this:

```
ifmx7: libinformix7.a informixEP.o informix.exp informix.imp
```

- Change the output name of the link-edit statement. For example, change this:

```
ld -o informix7 \
```

To this:

```
ld -o ifmx7 \
```

The name will be used in the `SERVER_PROTOCOL` column of `SYSCAT.SERVERS` for the first database that is accessed using Informix

Version 7. The SERVER_PROTOCOL column has a restriction that the filename cannot be an existing data access module name unless you are replacing it.

- c. As root, link-edit the informix7 data access module library using the supplied makefile. To complete this task, you need the following:
 - The informix7 data access module library that is provided with DataJoiner
 - The informix.exp file that is provided with DataJoiner
 - The Informix esql/rt libraries that are provided with the Informix esql product

From the /usr/lpp/djx_02_01_01/lib directory, execute the makefile by entering one of the following commands:

- If using ESQL/RT, enter:

```
make -f djxlink.makefile informix7
```
- If using ESQL/C, enter:

```
make -f djxlink.makefile informix7c
```

If you changed the name of the data access module in the makefile, then use that name with the **make** command.

Whenever you re-run djxlink.makefile for the informix data access module, you must stop and restart all DataJoiner instances that use the informix7 data access module.

Complete steps 4 through 7 on page 140 for each DataJoiner instance that will use the informix data access module.

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 dbserver 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"
```

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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

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

Where *informix-home-directory* is the directory in which you installed the e/sql library in step 1 on page 137, 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.

Be sure there are no spaces on either side of the equal sign (=) in these export statements. 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 db2profile 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 have a data access module that is named *informix7* and another data access module that is named *ifx7*, the entry would be:

```
export DJXCOMM=informix7,ifx7
```

See "Step 2: Set Environment Variables" on page 57 for more information.

7. Invoke db2profile through either of the following methods:
 - From the DataJoiner instance login, run .profile. For example:


```
. .profile
```
 - Run it directly. For example:


```
./home/djinst1/sql1lib/db2profile
```
8. 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 infmxb  
SETTING 'I'
```

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

9. 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.

10. 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 8 on page 140. 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 10. Accessing Microsoft SQL Server and Sybase Data Sources Using Sybase Open Client

The Sybase data access module is included with DataJoiner for AIX and supports both the ctlib and dblib access mechanisms.

You can also use the Sybase Open Client dblib API to access Microsoft SQL Server.

Follow these steps:

1. Install catalog stored procedures on the Sybase or Microsoft SQL Server 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.
3. If not already completed as part of install, use `djxlink.sh`, as described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47, to link-edit Sybase Open Client libraries with DataJoiner.

Complete Steps 4 through 7 for each DataJoiner instance that will use the Sybase data access module.

4. Update the `.profile` file of the DataJoiner instance with the SYBASE environment variable. Perform an export with the following commands:

```
export SYBASE="sybase home directory"
export PATH="$SYBASE/bin:$PATH"
```

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

5. Execute the DataJoiner instance `.profile` by entering:
`. .profile`
6. Recycle (**db2stop**, then **db2start**) the DataJoiner instance.
7. Use the Sybase-supplied utility to create an interfaces file that includes the data for all the Sybase Open Servers and Microsoft SQL Servers you want to access. See the *Sybase System Administration Guide* for more information on this procedure. Figure 32 on page 144 contains an example of a Sybase Open Client interfaces file:

Sybase Open-Client interfaces file

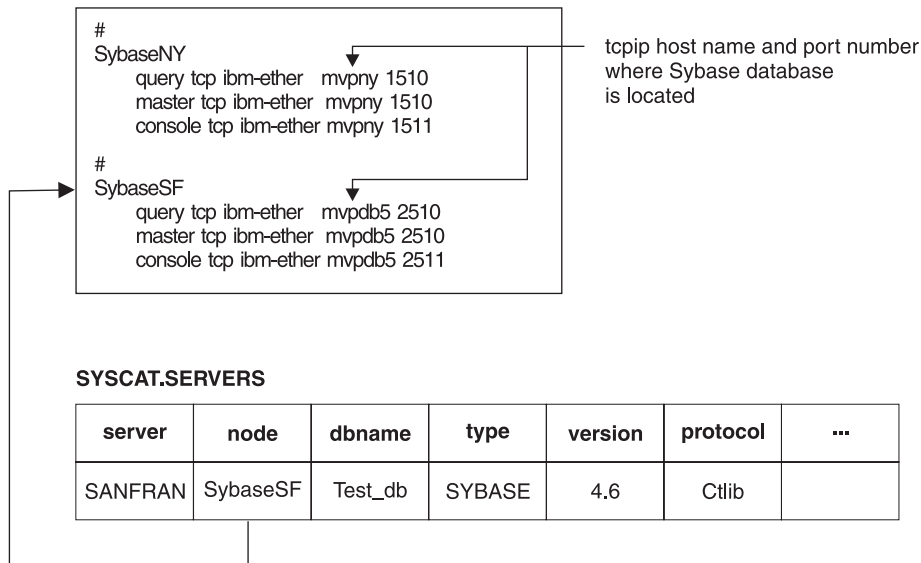


Figure 32. Sybase Data Access Module. The database server name in the Open Client interfaces file corresponds to the `NODE` column in `SYSCAT.SERVERS`; the host machine name and port number correspond to the TCP/IP host machine name and port address for the Sybase data source.

8. Make the interfaces file accessible by doing *one* of the following:
 - Put the file in the DataJoiner instance's `$HOME/sqllibdirectory`.
 - Using the `In` command, link to the file from the DataJoiner instance's `$HOME/sqllib` directory. For example,


```
In -s -f /home/sybase/interfaces /home/djinst1/sqllib
```
 - 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 146.
9. Use the `CREATE SERVER MAPPING DDL` statement to define each database you wish 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 32. 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`:
 - For Sybase data sources, specify `sybase`.

- For Microsoft SQL Server data sources, specify MSSQLSERVER.
- Set VERSION to the data source version that the server is running. For example, 4.3 for Sybase 4.3.
- Set PROTOCOL:
 - For Sybase data sources, specify either "ctlib" or "dblib"
 - For Microsoft SQL Server data sources, specify "dblib"

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"
```

A sample entry for Microsoft SQL Server Version 6 is:

```
CREATE SERVER MAPPING FROM MsoftV6 TO NODE "MsoftV6" DATABASE "test_db"
TYPE MSSQLServer VERSION 6.0 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.

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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 mssqlsoft.\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

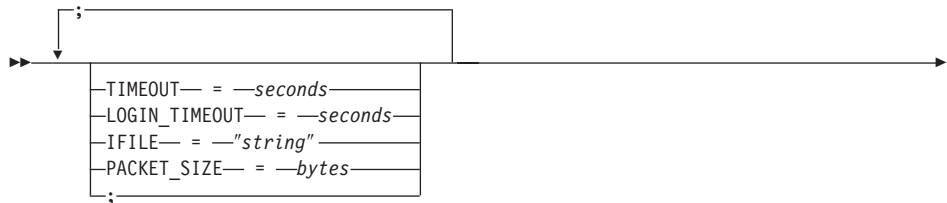
Open Client uses timeout thresholds to interrupt queries that run for too long a period of time. You can set these thresholds in DataJoiner by using the CONNECTSTRING option of the CREATE SERVER OPTION DDL statement.

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 dblib is 0, and for ctplib is -9999. Both of these default values cause 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 sqllib/interfaces 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 /etc/interfaces, set the connect string using the following example:

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

Chapter 11. Accessing Microsoft SQL Server through the Visigenic ODBC Driver

Follow these steps if you plan to access to Microsoft SQL Server through the Visigenic ODBC driver:

1. If you have not already done so, install the Visigenic ODBC driver on the system where DataJoiner is installed. See the Visigenic *ODBC DriverSet Installation and Configuration Guide* for more information.
2. If you have not already done so, build a Visigenic data access module by running `djxlink.sh`, as described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

For DataJoiner to access Microsoft SQL Server databases through the Visigenic ODBC driver, DataJoiner needs a data access module, which essentially is a file that has been built by link-editing DataJoiner libraries with Visigenic libraries.

When you run `djxlink.sh`, if you receive a message that indicates that the Visigenic data access module could not be built, you will need to edit the sample makefile named `/usr/lpp/djx_02_01_01/lib/djxlink.makefile` so that the makefile works with your particular setup. Editing and running `djxlink.makefile` are also described in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.

If `djxlink.sh` fails to build a Visigenic data access module, login as root and edit `djxlink.makefile`. Find the section in the file that is titled “Accessing Data Sources Through Visigenic ODBC Drivers.”

- a. In the makefile, change the `DJX_ODBC_LIBRARY_PATH` variable to the directory that contains the Visigenic libraries.
- b. The name of the Visigenic data access module that is built by the makefile is `mssqlodbc`. If you do not want to change this name, skip to the next step.

If you want to change the name, you need to change two lines in the makefile:

- Change the target name on the dependency line. For example, if you want the data access module to be named `visiodbc`, change the dependency line from this:

```
mssqlodbc: djxmssql.exp libdb2e.exp libmssql.a
```

To this:

```
visiodbc: djxmssql.exp libdb2e.exp libmssql.a
```

- Change the output name of the link-edit statement. For example, change this

```
ld -o mssqlodbc \
```

To this:

```
ld -o visiodbc \
```

- c. Link-edit the data access module by using the supplied makefile. To complete this task, you need the following:
 - The `libmssql.a` data access module library that is provided with DataJoiner

- The djxmssql.exp file that is provided with DataJoiner
- The Visigenic libraries that are provided with the Visigenic ODBC Drivers

Run the makefile by entering:

```
cd /usr/lpp/djx_02_01_01/lib
make -f djxlink.makefile mssqlodbc
```

If you changed the name of the data access module in the makefile, then use that name with the **make** command. For example:

```
make -f djxlink.makefile visiodbc
```

Whenever you rerun djxlink.makefile for the Visigenic data access module, you must stop and restart all DataJoiner instances that use the Visigenic data access module.

Complete Steps 3 through 11 on page 152 for each DataJoiner instance that will use the Visigenic data access module.

3. If you want the Visigenic data access module to start dynamically when the DataJoiner instance is started, update the db2profile of the DataJoiner instance to include an entry for the DJXCOMM variable. The name of each unique Visigenic data access module you created should be specified. If you do not set the DJXCOMM variable, you will need to manually start each Visigenic data access module.

For example, if you have a data access module that is named mssqlodbc and another that is named visiodbc, the entry would be:

```
export DJXCOMM=mssqlodbc,visiodbc
```

See “Step 2: Set Environment Variables” on page 57 for more information.

4. Invoke db2profile by doing one of the following:
 - From the DataJoiner instance login, run .profile; for example:


```
. .profile
```
 - Run it directly; for example:


```
./home/djinst1/sqllib/db2profile
```
5. Use the Visigenic-supplied odbc.ini file to define your Microsoft SQL Servers to the Visigenic ODBC driver. See your Visigenic documentation for more information on this procedure. The following figure shows the general structure of an odbc.ini file.

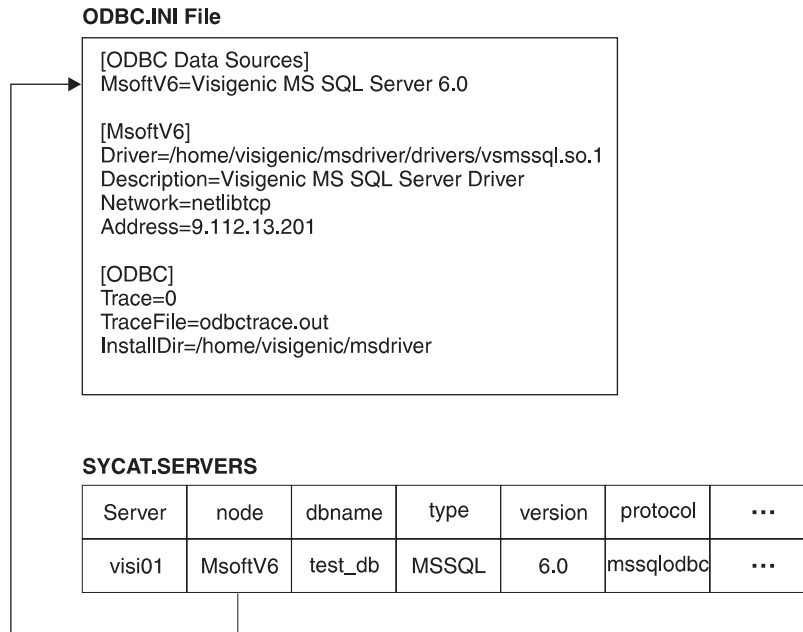


Figure 33. Visigenic ODBC Data Access Module

6. Recycle (**db2stop** followed by **db2start**) the DataJoiner instance.
7. Make the `odbc.ini` file accessible to your DataJoiner instance by doing one of the following:
 - Put the file in the DataJoiner instance's `$HOME` directory.
 - Using the `ln` command, make a symbolic link to the file from the DataJoiner instance's `$HOME` directory. For example,

```
ln -s -f /home/visigenic/odbc.ini /home/djinst1/.odbc.ini
```
8. Using the `ln` command, make a symbolic link from the following libraries to your `sqllib/lib` directory:
 - `libodbcinst.a`
 - `libodbc.a`

You must do this for each DataJoiner instance.

9. Use the `CREATE SERVER MAPPING DDL` statement to define each Microsoft SQL Server data source that is defined in the `odbc.ini` file. When performing server mapping:
 - Choose a unique server name.
 - Set `NODE` to the appropriate Microsoft SQL Server data source name from the `odbc.ini` file. This value is case sensitive. For an example, see Figure 33.

Note that you do not need to use the **CATALOG NODE** command for data sources that are accessed through Visigenic.

- Set DATABASE to the name of a database at the Microsoft SQL Server that is listed in the NODE column. This value is case sensitive.
- Set TYPE to MSSQLSERVER.
- Set VERSION to the version of Microsoft SQL Server. For example, 6.0 for Microsoft SQL Server 6.0.
- Set PROTOCOL to the name of the data access module you built in 2 on page 149 . For example, mssqlodbc. This value is case sensitive.

A sample entry for Microsoft SQL Server Version 6 is:

```
CREATE SERVER MAPPING FROM MsoftV6 TO NODE "MsoftV6" DATABASE "test_db"
TYPE MSSQLServer VERSION 6.0 PROTOCOL "mssqlodbc"
```

In this example, the server is the same as the node. The NODE value must match the data source name in the odbc.ini file.

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

10. 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 Microsoft SQL Server are:

```
CREATE SERVER OPTION colseq FOR SERVER MsoftV6
SETTING 'I'

CREATE USER MAPPING FROM marlow TO SERVER MsoftV6
AUTHID "bigdog" PASSWORD "growler"
```

11. Create nicknames. When you create nicknames, remember to put double quotes around the Microsoft SQL Server schema and table name as they are case sensitive. For example:

```
CREATE NICKNAME sanjose.employees FOR MsoftV6."st1"."employees"
```

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

12. 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 10 on page 152. 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 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.

Restriction:

Oracle's Net8 client software is supported on AIX 4.1.5 and higher only.

To set up access to Oracle databases:

1. Install Oracle SQL*Net or Net8 on the DataJoiner system.
It is recommended that you request a relink of SQL*Net or Net8 during the install process. This information will be used when DataJoiner is linked to this software.
2. If not already completed as part of install, use `djxlink.sh` as described in "Step 3: Link DataJoiner to Data Source Client Software" on page 47 to link-edit Oracle SQL*Net or Net8 libraries to DataJoiner.
3. Configure SQL*Net or Net8. See the Oracle documentation for configuration instructions.

Complete Step 4 through Step 8 on page 156 for each DataJoiner instance that will use the Oracle data access module.

4. If not already done, Set the `ORACLE_HOME` environment variable. 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.

Set environment variables in the DataJoiner instance owner's profile by entering the following commands:

```
export ORACLE_HOME=oracle-home-directory
export ORACLE_BASE=oracle-home-directory
```

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

`ORACLE_BASE` is optional. It is used to locate the `tnsnames.ora` file.

5. Ensure that the SQL*Net or Net8 `tnsnames.ora` file (usually found in the `/etc` directory) 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.

6. 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.

Set the TNS_ADMIN environment variable in the DataJoiner instance owner's profile by entering the following command:

```
export TNS_ADMIN=$ORACLE_BASE/admin/util/network
```

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

7. If you are using Oracle 7.2, set the ORA_NLS environment variable in the DataJoiner instance owner's profile by entering the following command:

```
export ORA_NLS=$ORACLE_HOME/ocommon/nls/admin/data
```

8. Recycle (**db2stop**, then **db2start**) the DataJoiner instance.
9. 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 34 on page 157.

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 "ora1a" 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.

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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 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 34 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.

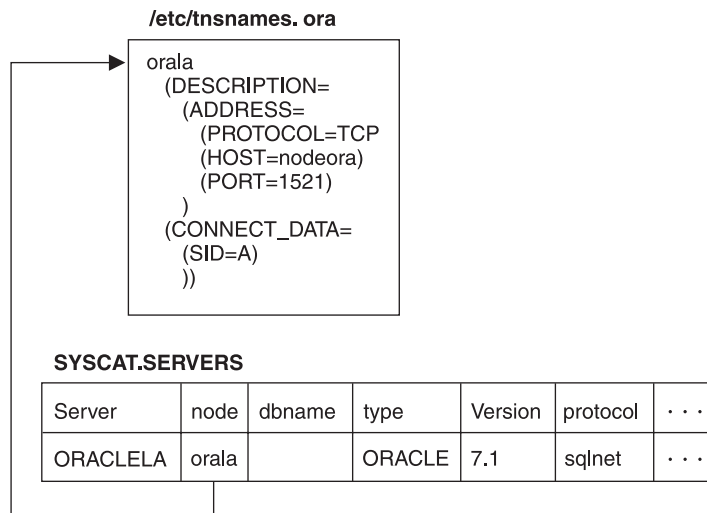


Figure 34. ORACLE Data Access Module

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

Whether you must update /etc/hosts 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 `/etc/hosts`. Otherwise, you need an entry for the remote host. See your network administrator to learn how your network is configured.

12. 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 10 on page 156. 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 Data Sources through CrossAccess

DataJoiner can access IMS, VSAM, or other data sources that are supported by CrossAccess (DataJoiner can also access IMS and VSAM data through Classic Connect, a separately orderable component of DataJoiner; refer to the *DataJoiner Classic Connect Planning, Installation, and Configuration Guide* for more information). This section explains how to configure DataJoiner to access IMS and VSAM through CrossAccess. Examples are specific for CrossAccess Version 2 Release 1. Refer to the CrossAccess documentation for detailed information on configuring CrossAccess for all data sources.

With DataJoiner already installed, follow these steps to configure a CrossAccess data access module. These instructions reference Figure 35 on page 163.

1. Install and configure the CrossAccess AIX components on the same system where DataJoiner is installed. After configuration is complete, note the path name of the CrossAccess dynamic link library directory and the name (complete path and filename) of the CrossAccess application configuration file. For example, if you modified the sample application configuration file that came with CrossAccess and plan to use it with DataJoiner, the Version 2 names you need to note are:

```
/usr/CrossAccess/UAV2R1M00/lib  
/usr/CrossAccess/UAV2R1M00/samples/cxa-run.cfg
```

The first name is the CrossAccess dynamic link directory; the second is the sample application configuration file.

2. Before using CrossAccess with DataJoiner, use CrossAccess's sample programs to test your configuration to CrossAccess. See the CrossAccess publications for information about running CrossAccess applications.
3. If you have not already done so, build a CrossAccess data access module by running `djxlink.sh`, as described in "Step 3: Link DataJoiner to Data Source Client Software" on page 47.

For DataJoiner to send requests to CrossAccess, it needs a data access module, which essentially is a file that has been built by link-editing DataJoiner libraries with CrossAccess libraries.

After running `djxlink.sh`, you will receive a message indicating whether or not the CrossAccess data access module was successfully built. If the data access module could not be built, you will need to edit the sample makefile named `/usr/lpp/djx_02_01_01/lib/djxlink.makefile` so that the makefile works with your particular setup. Editing and running `djxlink.makefile` are also described in "Step 3: Link DataJoiner to Data Source Client Software" on page 47.

If `djxlink.sh` fails to build a CrossAccess data access module, login as root and edit `djxlink.makefile`. Find the section of the file entitled "Accessing Data Sources Through CrossAccess."

- a. "Xaccess" is the name of the CrossAccess data access module that is built by the makefile. If you do not want to change this name, skip to the next step.

If you want to change the name, you need to change two lines in the makefile:

- Change the target name on the dependency line. For example, if you want the data access module to be named "cxav1r2," change the dependency line from this:

```
Xaccess: djxxaccess.a Xaccess.exp Xaccess.imp
```

to this:

```
cxav1r2: djxxaccess.a Xaccess.exp Xaccess.imp
```

- Change the output name of the link-edit statement. For example, change this:

```
ld -o Xaccess \
```

to this:

```
ld -o cxav1r2 \
```

This name will be used in the SERVER_PROTOCOL column of SYSCAT.SERVERS in a later step. The SERVER_PROTOCOL column has a restriction that the filename cannot be an existing data access module name unless you are replacing it.

- If the CrossAccess dynamic link library directory on your system is not the same as the one specified with the -L option in the makefile, replace the -L path with your own. For example, if your path is /usr/lpp/CrossAccess/lib, change this:

```
-L/usr/CrossAccess/UAV2R1M00/lib
```

to this:

```
-L/usr/lpp/CrossAccess/lib
```

- Ensure that the reference to the apif.imp includes the CrossAccess directory where the file is located. This will ensure that the proper level is referenced during the link-edit.

```
-bI:/usr/CrossAccess/UAV2R1M00/samples/apif.imp
```

- As root, link-edit the CrossAccess data access module library with the CrossAccess client libraries, using the supplied makefile.

Ensure that you are in the DataJoiner "lib" directory and execute the makefile as follows:

```
cd /usr/lpp/djx_02_01_01/lib
make -f djxlink.makefile Xaccess
```

If you changed the name of the data access module in the makefile, you need to use that name with the **make** command. For example:

```
make -f djxlink.makefile cxav1r2
```

Whenever you rerun djxlink.makefile for the CrossAccess data access module, you must recycle all DataJoiner instances that use the CrossAccess data access module.

4. For each DataJoiner instance that needs to work with CrossAccess, set the CrossAccess CXA_CONFIG environment variable in the DataJoiner instance's .profile file. Set it to the name of the CrossAccess application configuration file you

made when you installed and configured CrossAccess. Using the example filename from Step 1, add the following line to each applicable DataJoiner instance .profile file:

```
export CXA_CONFIG=/usr/CrossAccess/UAV2R1M00/samples/cxa-run.cfg
```

Note: If you have more than one CrossAccess instance on a given system, make sure that CXA_CONFIG points to the correct application configuration file for the CrossAccess instance the DataJoiner instance should use.

5. Rerun the DataJoiner instance owner's profile. Then recycle the DataJoiner instance. From the instance owner's home directory:

```
. .profile
db2stop
db2start
```

6. Update the catalog views by using the CREATE SERVER MAPPING statement to define each IMS or VSAM data source. Each statement corresponds to a CrossAccess SYSLOCATION configuration parameter. When performing server mapping:

- Choose a unique server name. It is DataJoiner's name for the particular IMS or VSAM data source. In this example XACCESS1 and XACCESS2 will be used for the VSAM and IMS data sources respectively.
- Set the NODE to the location name in the CrossAccess SYSLOCATION configuration parameter. In this example, SFIMS and SFVSAM are used for the IMS and VSAM data sources respectively. This value is case sensitive.
- Always use xaccess or crossaccess for the TYPE.
- Set VERSION to the version of CrossAccess, 2.1.
- The unique filename specified in Step 3, Xaccess, is used as the PROTOCOL. This value is case sensitive

A DATABASE value is not required; neither IMS nor VSAM has multiple databases per instance to which you must connect to access data.

A sample entry for an IMS data source is:

```
CREATE SERVER MAPPING FROM imsprod TO NODE "sfims"
TYPE xaccess VERSION 2.1 PROTOCOL "Xaccess"
```

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

If desired, perform user mapping. Specify a remote authid and password for any of the data sources by using the CREATE USER MAPPING statement. This is only necessary to send values that differ from the DataJoiner client userid.

After updating the catalog views, all users must reconnect to the DataJoiner database to access the new servers.

7. When creating nicknames for tables to be accessed in VSAM or IMS, use the following values for the remote-object-name:
 - The *data-source-name* corresponds to the SERVER column in the entry in SYSCAT.SERVERS that is specified in Step 6.
 - The **authorization-name** and the *remote-table-name* correspond to the table name parameter in the CrossAccess USE TABLE 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 6 on page 161 . 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.

Figure 35 on page 163 shows the configuration for the CrossAccess data access module configuration within DataJoiner.

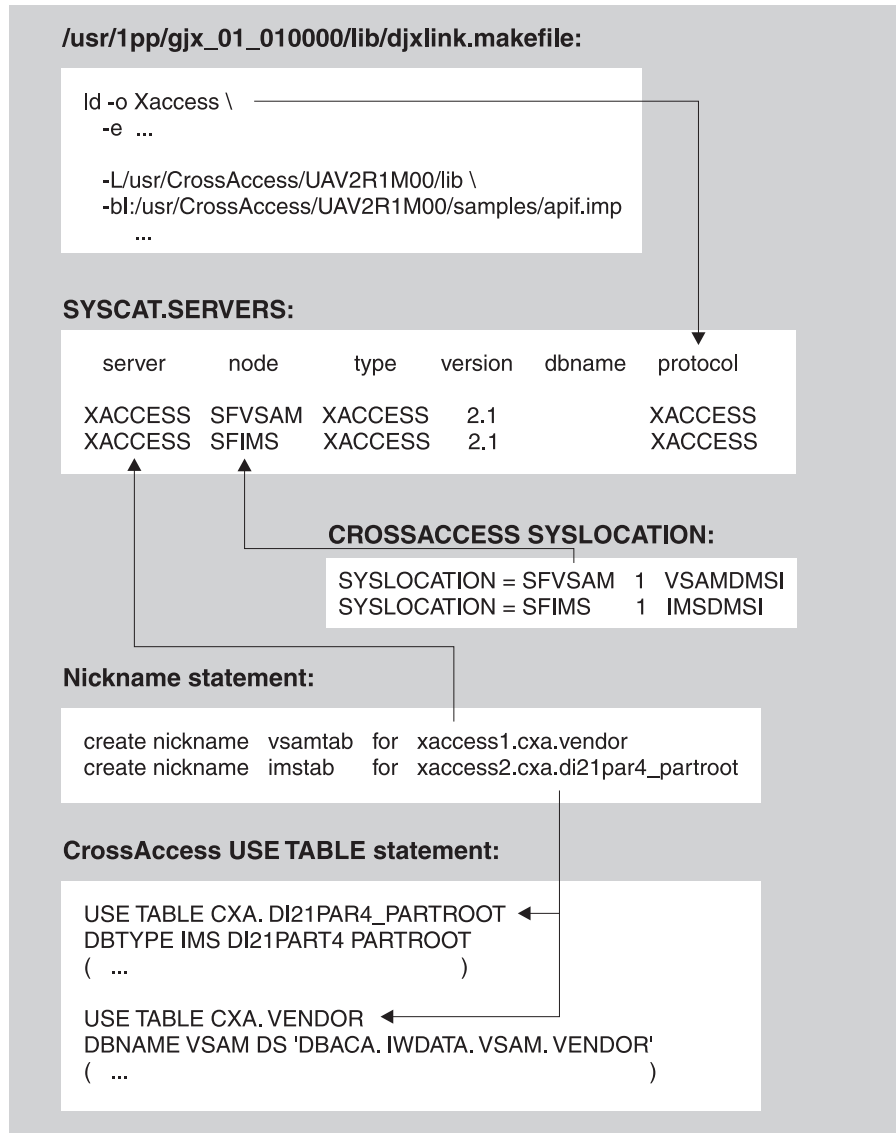


Figure 35. System Catalog Tables

Chapter 14. 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.

With DataJoiner installed, complete the following steps:

1. Install the code that contains the ODBC or X/OPEN-compliant driver on the DataJoiner system.
2. As root, edit the supplied makefile (`/usr/lpp/djx_02_01_01/lib/djxlink.makefile`) to specify the library directory of the driver you installed in Step 1 and a unique file name for the output of the link-edit. The unique file name will be used in the `SERVER_PROTOCOL` column of `SYSCAT.SERVERS`. The filename cannot be an existing data access module name unless you are replacing it.

To link-edit the custom data access module, you need:

- The custom data access module library that is provided with DataJoiner
- The custom data access module exports file that is provided with DataJoiner
- The custom data access module imports file that is provided with DataJoiner
- The CLI libraries that are provided by CLI vendor

You need to perform the link-edit for each CLI driver (each distinct access protocol) you want to use. For example, if you have an X/OPEN-compliant driver to get to an RDB, you need to perform the link-edit once regardless of the number of RDBs you want to access. If you have both an X/OPEN-compliant driver to get to the RDB and a different X/OPEN-compliant driver to get to Supra, you need to link-edit twice—once for the RDB driver and once for the Supra driver. You can then configure access to as many RDB or Supra systems as you want without additional link-editing.

To execute the makefile, enter:

```
cd /usr/lpp/djx_02_01_01/lib
make -f djxlink.makefile Generic
```

Whenever you rerun `djxlink.makefile` for the generic data access module, you must recycle all DataJoiner instances that use the generic data access module.

Complete the remaining steps for each DataJoiner instance that will use the generic data access module.

3. If your driver needs specific environment variables set, identify the variables to a given DataJoiner instance before you use the driver with that instance. To accomplish this task, you must add an entry to the `sqllib/.envfile` of the instance. Each entry in `sqllib/.envfile`, except the last entry, has the following format:
`variable_name+`

Do not put the plus sign (+) on the last entry. For example, the contents of the sqllib/.envfile might look like this:

```
HOME+
SYBASE+
CXA_CONFIG+
ORACLE_HOME+
TNS_ADMIN
```

If you have a driver that needs an environment variable different from these, such as QE_ODBC_INI, you can add it. The file might look like this:

```
HOME+
SYBASE+
CXA_CONFIG+
ORACLE_HOME+
TNS_ADMIN+
QE_ODBC_INI
```

4. 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.
5. Recycle (**db2stop**, then **db2start**) the DataJoiner instance.
6. Issue a CREATE SERVER MAPPING DDL statement for each data source you added in step 2 on page 165.
 - 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"
```

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 UNIX operating system command prompt, you need to enclose all case-sensitive parameters in double quotes, and you need to enclose the entire CREATE SERVER MAPPING statement in single quotes.

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* 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"
```

8. 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.

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 166 .

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 15. Configuring for Two-Phase Commit

This chapter 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 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

Table 5. Data Sources Supporting Two-Phase Commit (continued)

Data Source	Supports two-phase commit?	Supports two-phase commit DDL?
SQL/DS, Version 4.1 and higher	Yes	Yes
Sybase SQL Server (CTLIB) with XA library	Yes	No
Sybase SQL Server (DBLIB)	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 ctplib) install the required XA libraries for the data source.

For Sybase ctplib 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 xa_config, 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, link-edit the XA libraries with the DataJoiner libraries by adding the following lines to the appropriate section of `djxlink.makefile` as follows:

Informix (7.1 and higher only)

Add `libinfxxa.o`

Oracle Add `-lxa`

Sybase (ctlib)

- Change `-l comn` to `-l comn.so`
- Change `-l cs` to `-l cs.so`
- Add `-lxa`

- 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 TCP/IP 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.

Part 4. Connecting Clients to DataJoiner

Chapter 16. Accessing Local and Remote Clients

Local clients require minimal configuration. For example, local clients do not require communication configuration. However, you must run the db2profile shell script as discussed in “Step 2: Set Environment Variables” on page 57.

You can either invoke the shell script directly or add it to a local client's .profile so that the database environment is established during login. You can also copy the script into a local client's directory structure and customize it.

A common approach is to change the setting of the DB2INSTANCE environment variable in a local client's db2profile. Multiple DataJoiner instances can be defined on a given system. DB2INSTANCE records the instance that a client wants to access. If a client has access to multiple DataJoiner instances, you must modify the value of DB2INSTANCE and rerun db2profile before you change instances.

Remote clients require either the DB2 Client Application Enabler (CAE) or the DB2 Software Developer's Kit (SDK) V2.1.1 or later to communicate with a DataJoiner server. Client access is available from any DB2 Family product that provides function equivalent to the DB2 CAE 2.1.1. DataJoiner for AIX supports both APPC (SNA) and TCP/IP clients. Please refer to your client documentation to determine the protocol to use to access DataJoiner.

Chapter 17. TCP/IP Support for Remote Clients

This section provides an overview of the process for configuring TCP/IP for both DataJoiner instances and clients and provides several examples. Refer to specific client documentation for more information on possible configurations.

For DataJoiner to provide communication support through TCP/IP, you must first install and configure the Base Operating System Network Facilities (bosnet), which is part of the AIX operating system.

Figure 36 shows a high-level configuration of both DataJoiner instances and client workstations.

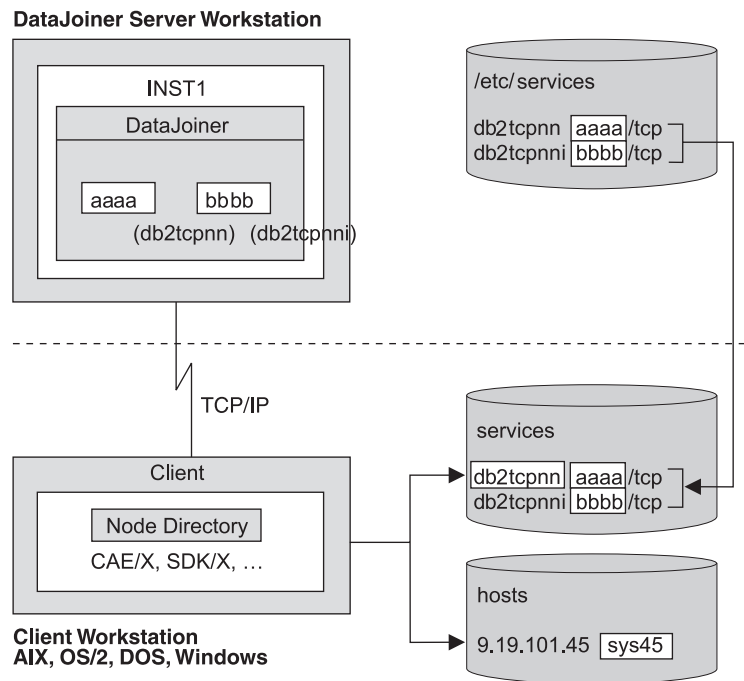


Figure 36. Remote TCP/IP Clients

Configuring DataJoiner Instances for TCP/IP Clients

To accept connection requests from clients that use TCP/IP, a DataJoiner instance must be listening on a port that is designated to that instance. A second port is used to handle interrupt requests from clients.

To configure these ports, the AIX system administrator must:

1. Define two `service_name` entries using two consecutive unique port numbers for every instance of DataJoiner on a workstation. The service names and associated ports are defined in the system's `/etc/services` file.

For example, if port 3750 is the main connection port, the entries in `/etc/services` are:

```
dbtcp01  3750/tcp
dbtcp01i 3751/tcp
```

The connection ports that are specified on the client must match the ports that are defined on the system where DataJoiner is installed. The main connection port (3750 in the example) must be a unique port number with a value greater than or equal to 1000. The interrupt port (3751 in the example) must be a unique port number equal to the value of the main port plus 1. The service names (`dbtcpnn` and `dbtcpnni` in the example) do not need to match the names on the client. However, using the same names is strongly recommended.

2. Specify the service name that is associated with the main connection port as the value for the `SVCENAME` parameter in the instance's database manager configuration file. The value of this parameter comes from the service name of the main port for the DataJoiner instance. When the instance starts, it uses this information to identify the ports to which it should listen. The interrupt port number is implied from the main connection port number and does not need to be explicitly identified in the database manager configuration file.

You can update the configuration file using the CLP, which is documented in the *DB2 Command and API Reference*. For example:

```
UPDATE DATABASE MANAGER CONFIGURATION USING SVCENAME dbtcpnn
```

where *dbtcpnn* represents the service name of the main connection port.

If you do not define the `svcename` parameter in the database manager configuration file, TCP/IP support will not be enabled when a **db2start** command is issued.

3. Set the `DB2COMM` environment variable to `TCPIP`. See "Chapter 20. Controlling Communication Support Initialization" on page 205 for more information.

Configuring the Client

To configure a TCP/IP client, you need to update the services file on the client machine and possibly the hosts file on the client machine.

You must add two entries to the services file. For example:

```
dbtcpnn  aaaa/tcp # DataJoiner main connection port
dbtcpnni bbbb/tcp # DataJoiner interrupt port
```

where *dbtcpnn* and *dbtcpnni* are any names not already in the services file. Your location might have a naming convention for the service names. In the example the names indicate a port for a database; *nn* is used to distinguish a particular instance.

The port numbers aaaa and bbbb must be decimal numbers greater than 1000. They must equal the port numbers that are defined on the system where DataJoiner is installed. The service names, dbtcpnn and dbtcpnni in the example, do not need to be the same on both systems (or all systems). However, using the same names is strongly recommended.

A hosts file is used to locate the system where DataJoiner is installed. It contains the DataJoiner system host name and address in addition to all the other host names and addresses. You do not need to use a hosts file if you either have a domain name server for the network or you code the IP address when you update the DataJoiner instance's database manager configuration file. Refer to a TCP/IP manual for details about domain name servers, and see your network administrator for information on whether the domain name server can be used instead of a hosts file.

To complete the TCP/IP configuration, a DataJoiner database must be cataloged on the client workstation, and client utilities must be bound at the DataJoiner database one time with appropriate authorities granted for all users. Refer to "Configuration Examples" to complete the configuration.

Tuning Network Options for Remote Failures Using 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 client application that accesses a remote DataJoiner server using TCP/IP, or the corresponding agent process at the server, might sometimes appear to be hung. You can use two network options, `tcp_keepidle` and `tcp_keepintvl`, to specify the 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 applications.

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

To remove an agent process at the server that is hung, use the **FORCE APPLICATION** command.

Configuration Examples

This section contains examples for configuring:

- A DataJoiner instance for TCP/IP clients
- Client Application Enabler (CAE) 2.1.1 for AIX or SDK 2.1.1 for AIX TCP/IP client. Although written for AIX, this example applies to any UNIX client that accesses DataJoiner.
- CAE 2.1.1 for OS/2 or SDK 2.1.1 for OS/2 TCP/IP client

For complete information on configuring CAE and SDK clients, refer to the documentation for the particular client.

The template in Figure 37 on page 181 is included for planning purposes. You can fill in the template with your client/server information and then complete the instructions in the appropriate example or steps in your client's publication.

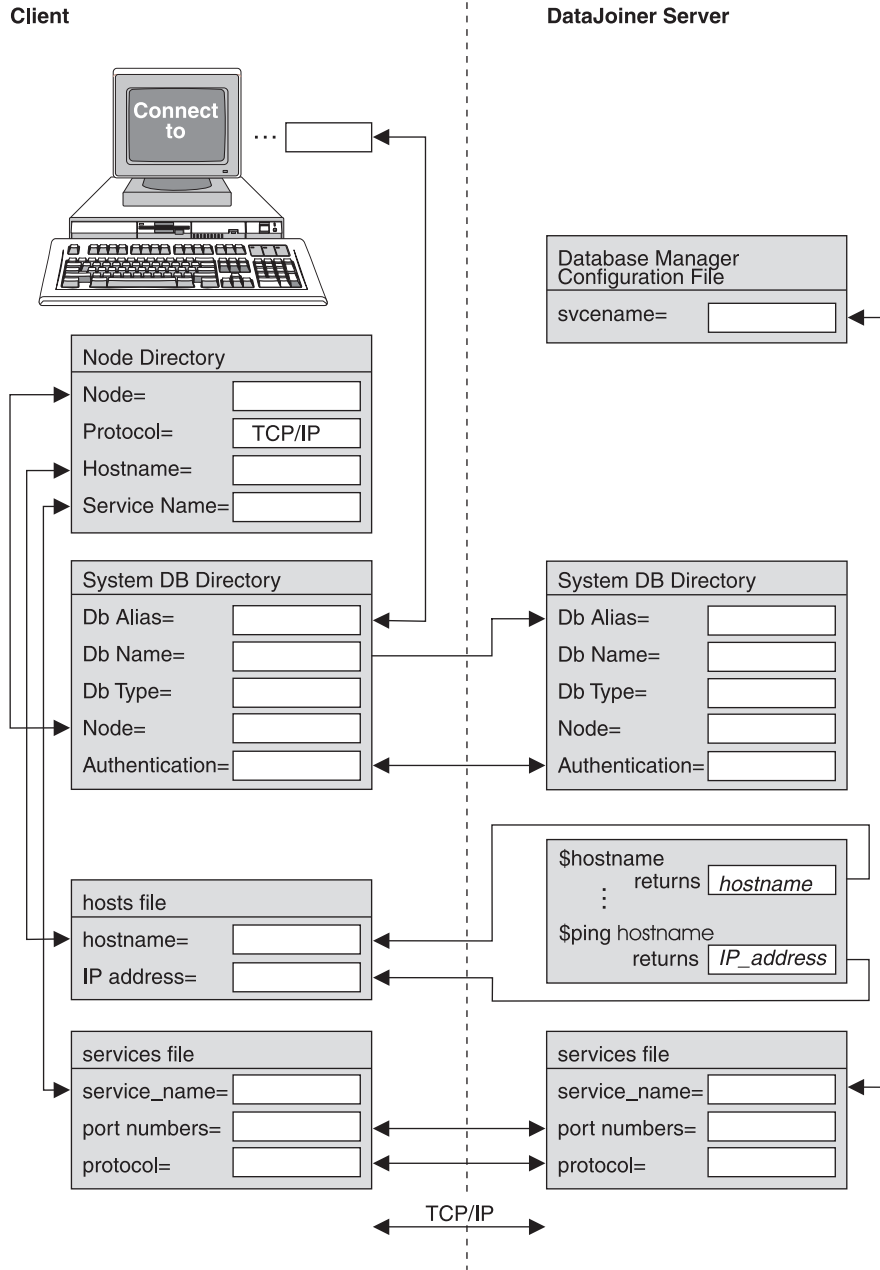


Figure 37. TCP/IP Planning Template

Configuring a DataJoiner Instance for TCP/IP Clients

For a DataJoiner instance to receive requests from a remote client as shown in later examples, you must configure it properly. Figure 38 shows how a DataJoiner instance is configured for a TCP/IP client.

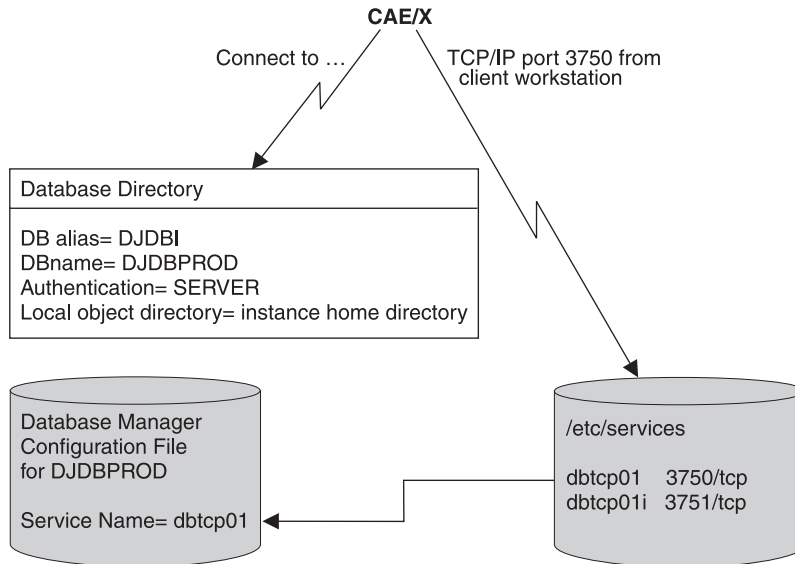


Figure 38. Configuring a DataJoiner Instance for TCP/IP Clients

Before configuring a DataJoiner instance for TCP/IP clients, note that:

- *dbtcp01* and *dbtcp01i* are any names not already in the */etc/services* file on the system where DataJoiner is installed.
- 3750 and 3751 are unique port numbers in the */etc/services* file.
- *djdb1* is the alias for the DataJoiner database. The database entry on the client must match this value.
- Authentication must equal SERVER or DCS if DB2/2 V1 clients are accessing DataJoiner. Other clients must have the same value as the DataJoiner database.

To configure a DataJoiner instance for TCP/IP clients:

1. As root user, add two entries to the */etc/services* file:

```
dbtcp01 3750/tcp # DataJoiner main connection port
dbtcp01i 3751/tcp # DataJoiner interrupt port
```

The port numbers 3750 and 3751 must be decimal numbers greater than 1000. The port numbers must be unique within the */etc/services* file. These values should have been used when the */etc/services* file was updated on the client.

2. Synchronize the `/etc/services` file and the `inetd` daemon by executing the following AIX commands:


```
inetimp
refresh -s inetd
```

This step is not required for AIX V4 or higher.
3. As the instance owner or as a system administrator, change the database manager configuration file by setting the service name value to the first port name in the `/etc/services` file with the following command:


```
db2 update database manager configuration using svcname dbtcp01
```
4. Issue the **db2start** command for this instance. Two listen processes start. The listen processes wait to be contacted by remote clients using the port numbers in the `/etc/services` file.
5. If the DataJoiner database was created with authentication set to server, add client logins to the workstation by using the appropriate operating system facilities.
6. Grant privileges on the DataJoiner database objects for each user as appropriate.

Configuring CAE 2.1.1 for AIX or SDK 2.1.1 for an AIX TCP/IP Client

The example in this section shows the TCP/P configuration for the CAE 2.1.1 AIX client. Figure 39 shows how configuration information is mapped to the `/etc/hosts` and `/etc/services` files. The process that is described here applies to all UNIX clients (AIX and HP-UX).

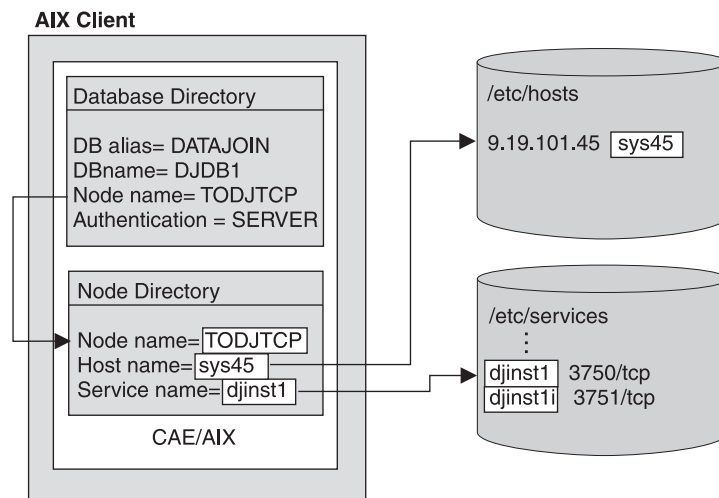


Figure 39. CAE 2.1.1 for AIX or SDK 2.1.1 for AIX TCP/IP Client Example

This example uses a hosts file because the node directory points to a host name and not the IP address.

To configure a CAE 2.1.1 for AIX TCP/IP client:

1. As root user on the AIX client, update the hosts and services file as follows:
 - a. Update the `/etc/services` file to define the same two port numbers that are used on the DataJoiner instance.
 - b. Synchronize the `/etc/services` file and the `inetd` daemon by executing the following AIX commands:

```
inetimp
refresh -s inetd
```

This step is not required for AIX V4 or higher.

- c. Update the hosts file, `/etc/hosts`, with the TCP/IP hostname of the DataJoiner machine. Depending on your network configuration, you might not need a hosts file. Contact your system administrator to learn about your network configuration.
 - d. Using `smitty` or the `mkuser` command, add any users to the system that need to connect to DataJoiner as clients from this system.

Before you complete Step 2 through Step 5 on page 185, note the following:

- `todjtcp` is the node entry name. This name appears in the system database directory and node directory.
- `djinst1` is the first port name in the `/etc/services` file.
- `sys45` is the host name of the system where DataJoiner is installed. Issue the `hostname` command on that system to obtain this value.
- `datajoin` is the alias name of the DataJoiner database.
- `djdb1` is the name of the DataJoiner database. It must match the alias name on the DataJoiner instance.

As the system administrator or the instance owner on the AIX client, complete the following tasks:

2. Catalog a node entry that points to the location of the remote database with the following command:

```
db2 catalog tcpip node todjtcp remote sys45 server djinst1
```

The `hostname` field in the node directory points to the TCP/IP host that contains DataJoiner. The `service name` field points to the DataJoiner instance to which the client is connecting.

3. Catalog the remote DataJoiner database in the system database directory with the following command:

```
db2 catalog database djdb1 as datajoin at
node todjtcp authentication server
```

4. If this is the first CAE 2.1.1 for AIX or SDK 2.1.1 for AIX client that is accessing a database, an auto-bind will be done if you have the appropriate privileges and authorization. Otherwise, you can bind the database manually. From an AIX prompt issue the following commands:

```
db2 connect to datajoin user username using password
db2 bind /sqllib/bnd/@db2ubind.lst blocking all grant public
```

- Grant connect privilege to any users who need access to the database.

Configuring CAE 2.1.1 for OS/2 or SDK 2.1.1 for an OS/2 TCP/IP Client

This section shows the TCP/IP configuration for CAE 2.1.1 for OS/2. Figure 40 shows how the configuration information is mapped.

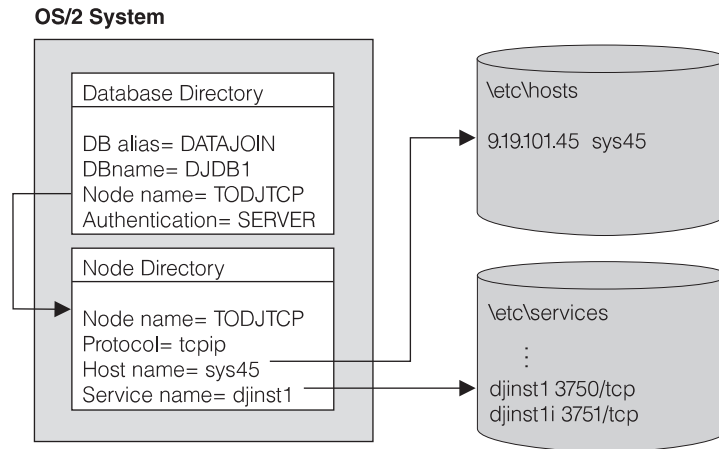


Figure 40. CAE 2.1.1 for OS/2 or SDK 2.1.1 for OS/2 TCP/IP Client Example

Configuring CAE 2.1.1 for OS/2 is similar to configuring CAE 2.1.1 for AIX. To configure a CAE 2.1.1 for OS/2 TCP/IP client:

- Configure for TCP/IP communications:
 - Update the `\etc\services` file to define the same two port numbers you used on the DataJoiner instance.
Your services file might be located in a directory different than the directory that is used in the example. To find the `\etc` directory for your machine, check the value of the `ETC` environment variable:

```
echo %ETC%
```
 - Optional: Update the local hosts file or a domain name server with the TCP/IP host name where DataJoiner is installed. The local hosts file is called `\etc\hosts`.
- From an OS/2 command line, log on as the OS/2 system administrator and add users who will connect to DataJoiner to UPM.

Before you complete the next steps, note the following:

- `todjtcp` is the node entry name. This name appears in the system database directory and node directory.
- `sys45` is the host name of the DataJoiner server or the IP address. Issue the **hostname** command on that system to obtain this value.

- *djinst1* is the name of the first port number in the `\etc\services` file.
 - *datajoin* is the alias name of the DataJoiner database.
 - *djdb1* is the name of the DataJoiner database. It must match the alias name on the DataJoiner instance.
3. From the OS/2 command line, catalog a node entry that points to the location of the remote database with the following command:

```
db2 catalog tcpip node todjtcp remote sys45 server djinst1
```

The `hostname` field in the node directory points to the TCP/IP host that contains DataJoiner. The `service name` field points to the DataJoiner instance to which the client is connecting.

4. Catalog the remote database in the system database directory with the following command:

```
db2 catalog database djdb1 as datajoin at node todjtcp  
authentication server
```
5. Bind the utility packages against the DataJoiner database if this is the first CAE 2.1.1 for OS/2 or SDK 2.1.1 for OS/2 client that accesses a database. From an OS/2 prompt, change to the `sqllib` directory and enter:

```
db2 connect to djdb1  
db2 bind @db2cliv1.lst messages bind.msg grant public
```
6. Grant connect privilege to any users who need access to the database.

Chapter 18. IPX/SPX Support for Remote Clients

DataJoiner provides IPX/SPX communication support through the IPX/SPX support option (db2_02_01.cs.ipx).

To enable IPX/SPX support on a 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 DataJoiner

IPX/SPX support is provided by:

- IBM Netware for AIX Version 3.11B (this is the complete product which includes a file server).

or

- IBM AIX NetBIOS and IPX Support/6000.

NetWare 4.x Directory Services is not supported in this release of DataJoiner. You should use the bindery emulation feature of NetWare 4.x.

You can access DataJoiner through IPX/SPX as follows:

- File Server Addressing.

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

- Direct Addressing.

The client connects to DataJoiner 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 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 DataJoiner 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 instance on a machine. It must also be unique among all IPX/SPX applications running on the DataJoiner machine to ensure that DataJoiner is able to listen to incoming IPX/SPX connections 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.

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

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

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

For example:

```
db2 update database manager configuration using fileservr 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 DataJoiner will be accessed by clients using only direct addressing.

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

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

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

Registering the DB2 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.



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 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:

1. This command must be issued locally from a DataJoiner server (it is not supported remotely) *once* before the first **db2start** for each DataJoiner instance.
2. If you move DataJoiner 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.

Chapter 19. APPC Support for Remote Clients

DataJoiner provides APPC support if you selected the SNA Clients option (dix_02_01_01.cs.sna) during DataJoiner install. To use APPC, SNA Server/6000 Version 2 must be installed and configured on the DataJoiner server workstation.

Remote clients require either the DB2 Client Application Enabler or the DB2 Software Developer's Kit to communicate with a DataJoiner server. Client access is available from any DB2 Family product that provides function equivalent to the DB2 CAE 2.1.1. DataJoiner for AIX supports both APPC (SNA) and TCP/IP clients. Refer to specific client documentation for more information on possible configurations.

There are different steps depending on the type of client you are configuring. This section includes two examples:

- "Configuring DataJoiner and AIX Clients"
- "Configuring DataJoiner and Configuring OS/2 Clients" on page 199

Configuring DataJoiner and AIX Clients

The example in this section identifies specific configurations for connecting a CAE for AIX client to a DataJoiner server by using APPC. Figure 41 on page 192 provides an overview of the detailed configuration that is used in this example. It is followed by Figure 42 on page 193, which you can use to define values for your configuration.

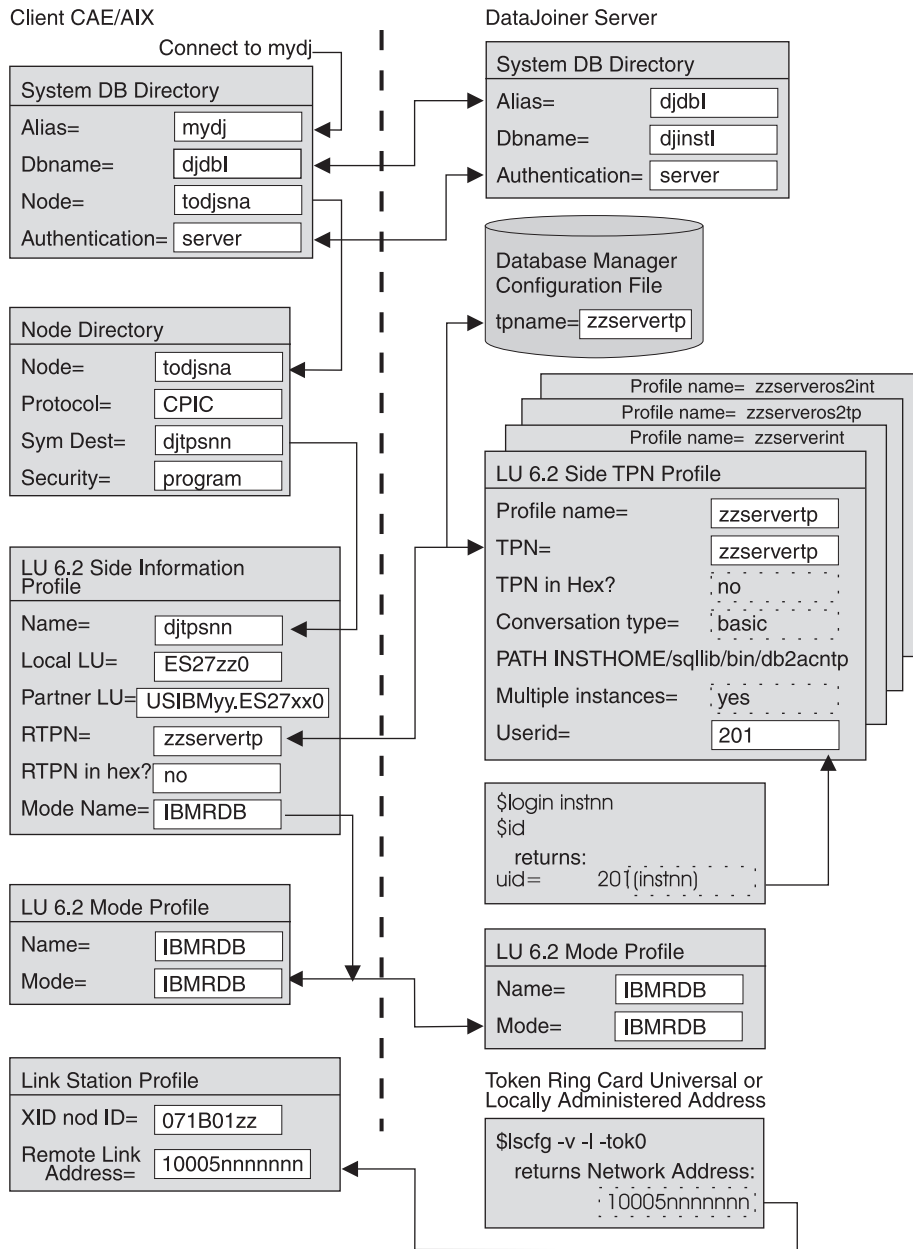


Figure 41. APPC Client Configuration

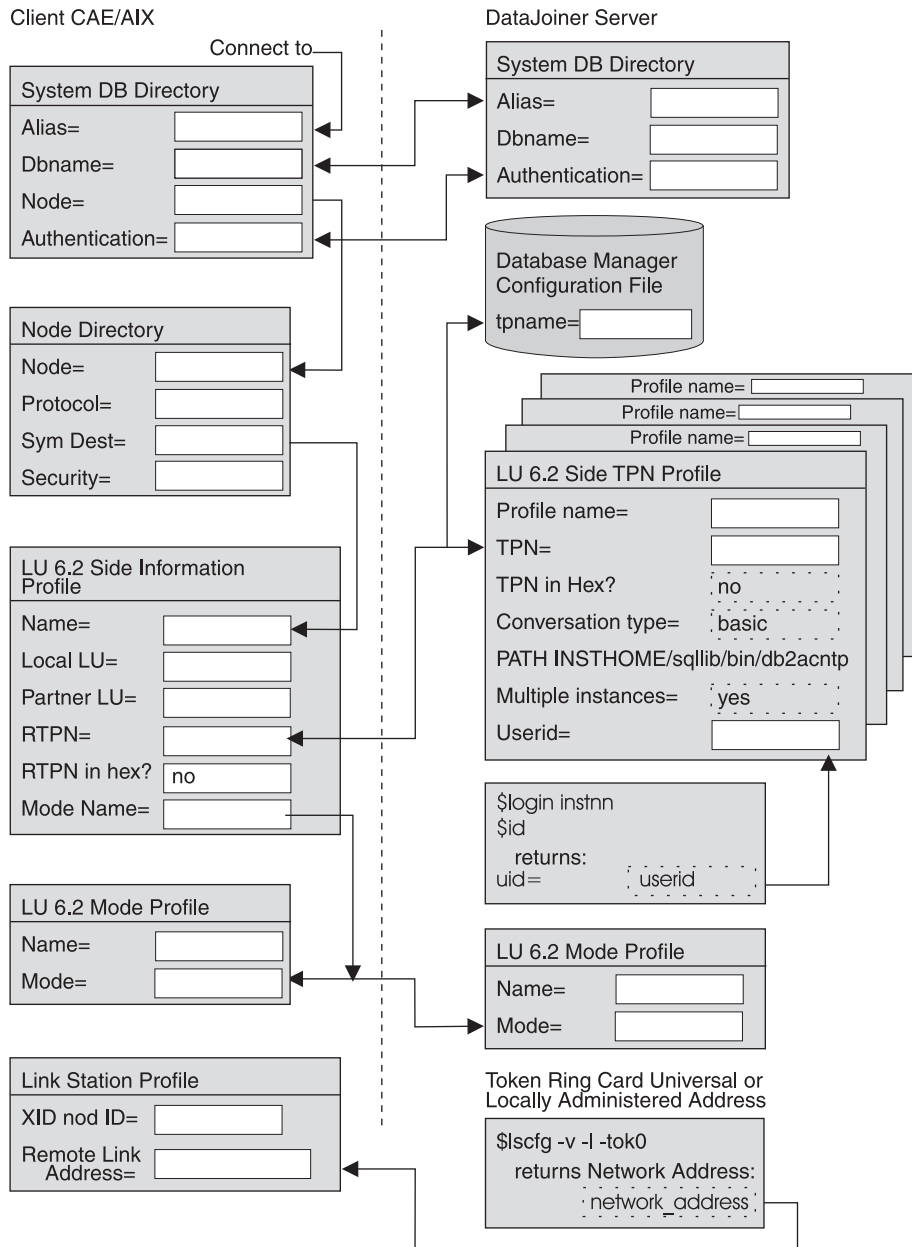


Figure 42. APPC Client Planning Template

Configuring DataJoiner for CAE for AIX APPC Clients

Complete the following steps to configure DataJoiner for CAE for AIX APPC clients:

1. Configure SNA Server/6000.
2. Update the database manager configuration.
3. Stop all connections.
4. Restart the instance.

Details for each step follow.

Step 1: Configure SNA Server/6000

To configure SNA Server/6000:

1. As root user, create AIX SNA Server/6000 profiles to define the SNA environment if they do not already exist. These profiles are:
 - Node profile (see Figure 17 on page 91)
 - Control point profile (see Figure 12 on page 82)
 - DLC profile (see Figure 10 on page 80)
 - Mode profile (see Figure 15 on page 87)
2. Create an LU 6.2 TPN profile. This TPN should match your database manager configuration file's tpname, which is set in "Step 2: Update the Database Manager Configuration" on page 197. Complete the following steps:
 - a. From the LU 6.2 menu, select **LU 6.2 Transaction Program Name (TPN)** and then select **Add a Profile**. You will see the window shown in Figure 43 on page 195 .


```

Change/Show LU 6.2 TPN Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...1]                                [Entry Fields]
New profile name                            []
Transaction program name (TPN)              [zzservertp]
Transaction program name (TPN) is in hexadecimal?  no          +
PIP data?                                   no          +
  If yes, Subfields (0-99)                  [0]         #
Use Command Line Parameters?               no          +
Command Line Parameters                     []
Conversation type                           basic       +
Sync level                                  none/confirm +
Resource security level                     none        +
  If access, Resource Security Access List Prof. []
Full path to TP executable                  <dj/sql1lib/bin/db2acntp]
Multiple instances supported?               yes         +
User ID                                     [221]      #
[MORE...9]

F1=Help          F2=Refresh      F3=Cancel      F4=List
F5=Reset         F6=Command      F7=Edit        F8=Image
F9=Shell         F10=Exit        Enter=Do

```

Figure 43. TPN Profile Screen

- b. Enter a profile name in the **New profile name** field. Consider using your actual TPN.
- c. Enter a transaction program name in the **Transaction program name (TPN)** field. You will need to enter this same name in the system management configuration file as the Transaction Program Name (TPNAME), which is described in “Step 2: Update the Database Manager Configuration” on page 197.
- d. Set the **Transaction program name is in HEXADECIMAL?** field to no.
- e. Set the **Conversation type** field to basic.
- f. Set the **Full PATH to TP executable** field to `inst/sql1lib/bin/db2acntp`, where `inst` is the home directory for the instance owner. For example, `/home/instdj`.
- g. Set the **Multiple instances supported?** field to yes.
- h. Set the **User ID** field to the instance owner’s user ID number (not the user name). You can obtain this value by entering the `id` command while logged on as the instance owner. You would get a result such as:


```
uid=221
```
- i. Use default values in the remaining fields.

- j. Press Enter to save the profile.
- k. Press F3 to return to the LU 6.2 Transaction Program Name menu.
3. Create an interrupt manager TPN profile by completing the following steps:
 - a. From the LU 6.2 Transaction Program Name menu, select **Add a Profile**. The **Change/Show LU 6.2 TPN Profile** window opens.

```

login
Change/Show LU 6.2 TPN Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...1]                               [Entry Fields]
New profile name                          []
Transaction program name (TPN)             [DB2INTERRUPT]
Transaction program name (TPN) is in hexadecimal?  no          +
PIP data?                                  no          +
  If yes, Subfields (0-99)                 [0]         #
Use Command Line Parameters?              no          +
Command Line Parameters                    []
Conversation type                          basic       +
Sync level                                 none/confirm +
Resource security level                    none        +
  If access, Resource Security Access List Prof. []
Full path to TP executable                 <dj/sqllib/bin/db2aittp]
Multiple instances supported?              yes         +
User ID                                    [221]      #
[MORE...9]

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command    F7=Edit     F8=Image
F9=Shell     F10=Exit     Enter=Do

```

Figure 44. TPN Profile for AIX Client

- b. Enter a profile name in the **New profile name** field. Consider using a name similar to the previous communication manager TPN name, such as `zzserverint`.
- c. Set the **Transaction program name (TPN)** field to `DB2INTERRUPT`.
- d. Set the **Transaction program name (TPN) is in HEXADECIMAL?** field to `no`.
- e. Set the **Full path to TP executable** field to `inst/sqllib/bin/db2aittp`, where `inst` is the home directory for the instance owner. For example, `/home/instdj`.
- f. Set the **Multiple instances supported?** field to `yes`.
- g. Set the **User ID** field to the instance owner's user name number (not the user name). You can obtain this value by entering the `id` command while logged on as the instance owner. You would get a result such as:

- ```
uid=221
```
- h. Use default values in the remaining fields.
  - i. Press Enter to save the profile.
  - j. Press F3 to return to the LU 6.2 Transaction Program Name menu.
4. If you plan to access DataJoiner from OS/2 clients, create two additional TPNs. See “Configuring DataJoiner and Configuring OS/2 Clients” on page 199 for more information.
  5. As root, start SNA Server/6000 and the listen attachment from smitty, or enter the following commands:

```
startsrc -s sna
startsrc -s l -p 'link_station_profile_name' &
```

## Step 2: Update the Database Manager Configuration

As the person with sysadm authority for the instance, use the following command to update the TPNAME field in the database manager configuration file. This name must match the LU 6.2 TPN name being used by the remote APPC client.

```
db2 update database manager configuration using tpname zzservertp
```

## Step 3: Stop All Connections

Stop all connections to the instance with either of the following commands:

```
db2 force applications all
db2stop
```

## Step 4: Restart the Instance

To restart the instance, enter the **db2start** command. Several listen processes start and wait to be contacted by remote clients using a transaction program name defined in a TPN profile.

## Configuring a CAE for AIX APPC Client

The steps in this section use the following values:

- *todjsna* is the node entry name. This name appears in the system database directory and node directory.
- *djdb1* is the database entry name. This name appears in the system directory on the client workstation and must match the alias entry on the DataJoiner server.
- *mydj* is the alias entry name in the system database directory on the client workstation.
- *server* is the authentication that is specified in the system directory on the client workstation.

As the person with sysadm authority on the client machine, complete the following steps to configure a CAE for AIX APPC client:

1. As root, enable the client for APPC communications by updating the SNA profile in smitty. Navigation steps within smitty are not shown here. These steps are only one way of configuring SNA and are not meant to be an exhaustive explanation. For more information, see the *AIX SNA Server/6000 Configuration Reference, SC31-7014*
  - a. Create a SNA DLC profile if one does not already exist. See Figure 10 on page 80 for an example.
  - b. Create a Link Station profile. The remote link address is the server's token-ring card universal or locally administered address. This profile contains the remote link address and the XID node ID. View the sample values below. You can see where they go by scrolling down in the window that is shown in Figure 11 on page 81 :
    - The remote link address is the DataJoiner server's token-ring card. This address can be obtained with the following command:

```
lscfg -v -1 tok0
```

which provides this result:

```
Network Address 10005AAB4169
```
    - The XID node ID is the IDBLK/IDNUM of the client machine in SNA Server/6000.
  - c. Create a control point profile if one does not already exist. The network ID is the NETID of the client machine. See Figure 12 on page 82 for an example.
  - d. Create a local LU profile for the client machine if one does not already exist. See Figure 13 on page 84 for an example.
  - e. Create an LU 6.2 Side Information Profile, where:
    - The profile name matches the symbolic destination name in the corresponding node directory entry on the client machine. See Figure 16 on page 89 for an example.
    - The local LU name is the LUNAME of the client machine.
    - The partner LU name is the LUNAME of the DataJoiner server machine. Alternatively, you can use the control point name in the control point profile.
    - The RTPN for CAE for AIX is any name that corresponds with an LU 6.2 TPN that points to the path INSTHOME/sqllib/bin/db2acntp on the DataJoiner server machine. This example uses zzservertp.
    - The mode name matches a mode name on the server machine.
  - f. Create an LU 6.2 Mode profile. The mode name in the profile should match the mode name that is defined in the LU 6.2 Side Information Profile on the client machine and the mode name that is defined on the DataJoiner server. See Figure 15 on page 87 for an example.
  - g. Verify the SNA profiles.
  - h. Start SNA Server/6000 with the following command:

```
sna -start sna
```

You can also use smitty to start SNA.

- i. Optional: Start the link station. The first time a client attempts to connect to the DataJoiner server, an attachment start request is automatically issued. If the link station was not started on the DataJoiner server, the link station times out and does not become active.

Start the link station with the following command:

```
sna -start 1 -p link_station_profile_name
```

You can also use smitty to start the link station.

2. Use the following command to create node entries that point to the location of the remote databases:

```
$ db2 catalog cpic node todjsna remote djtpsnn security program
```

3. Create remote database entries in the system database directory with the following command:

```
$ db2 catalog database djdbl as mydj at node todjsna authentication server
```

---

## Configuring DataJoiner and Configuring OS/2 Clients

OS/2 clients vary by type and version level. This section describes APPC client configuration to DataJoiner.

First follow the steps in “Configuring DataJoiner for CAE for AIX APPC Clients” on page 194 to configure DataJoiner for CAE for OS/2 or SDK for OS/2 Version 2 clients.

Then read “Configuring a DB2 for OS/2 Version 2 Client”. This section provides detailed information on configuring the OS/2 workstation for CAE and SDK Version 2 clients.

### Configuring a DB2 for OS/2 Version 2 Client

This section provides a detailed example for configuring the OS/2 workstation as a client to DataJoiner. The following steps are required to configure CAE for OS/2 and SDK for OS/2 V2 clients:

1. Update Communications Manager/2
2. Catalog the DataJoiner database
3. Connect to DataJoiner
4. Bind the database utilities

Details for each step follow.

## Step 1: Update Communications Manager/2

You need to customize various profiles at the client. The profiles that are required at the client to support communications with DataJoiner over a token-ring network are included in the following list:

- Local Node Characteristics
- Token Ring or Other LAN Types DLC Adapter Parameters
- Adapter List
- Connection to a Peer Node
- Partner LUs
- CPI Communications Side Information
- Mode Definition

The instructions show how to modify or add a profile by using Communications Manager Version 1.11. The steps are slightly different if you are using Version 1.1.

To reach the appropriate window for each profile, select options in the following order:

1. Double-click the **Communication Manager/2** icon.
2. Double-click the **Communication Manager Setup** icon.
3. Click the **Setup** push button. The Open Configuration panel appears.
4. Enter a new configuration name or select an existing configuration file. Click the **OK** push button. The Communications Manager Configuration Definition window opens.
5. From the **Options** menu, select **Configure any profile or feature**. The Communications Manager Profile List window opens.

You can add a new profile or modify an existing profile. The procedures for each of the sections in this step begin from the Communication Manager Profile List window.

**Local Node Characteristics Profile:** The local node characteristics profile contains the parameters that are needed to identify the local workstation to the SNA network. There should be one profile per workstation.

To update the Local Node Characteristics Profile:

1. On the Communications Manager Profile List window, select **SNA local node characteristics** and click the **Configure** push button.
2. For example, use the following values:

**Network ID**

J93DB2

**Local node name**

WSD2185

This is the name other workstations know this OS/2 workstation by. It must be unique in the network.

**Node type**

End node

**Local node ID (hex)**

05D D2185

If the workstation is connecting to a host through a 37x5 controller, you must enter this value (it is the IDBLK and IDNUM values concatenated). Otherwise, it is optional.

3. Click **OK** when done.

**Token Ring or Other LAN Types DLC Adapter Parameters Profile:** This profile is also called the LAN DLC Adapter Profile. It identifies the type of LAN adapter and adapter characteristics. There is one profile per adapter, and typically, there is only one per machine.

To update the Token Ring or Other LAN Types DLC Adapter Parameters Profile:

1. On the Communications Manager Profile List window, select **DLC - Token-ring or other LAN types** and click the **Configure** push button.
2. For example, use the following values:

**Adapter** 0**C&SM LAN ID** J93DB2

This value matches the network ID on the local node characteristics profile.

**All other values**

Default

3. Click **OK** when done.

**Adapter List Profile:** This profile sets the adapter you want to use for the connection. This example shows how to create a profile using the Token Ring adapter setup.

To create an Adapter List Profile:

1. On the Communications Manager Profile List panel, select **SNA connections** and click the **Configure** push button.
2. For partner type, select the **To peer node** radio button, and then select **Create**.
3. Be sure the adapter number matches the adapter on the Token Ring or Other LAN Types DLC Adapter Parameter profile. The example uses 0.
4. Select **Continue** when done.

**Connection to a Peer Node Profile:** This profile defines the connection from your workstation to the peer node. You can find the values to use on the DataJoiner workstation.

To create a Connection to a Peer Node Profile:

1. On the Communications Manager Profile List panel, enter the value that describes this link to DataJoiner, such as LINKDJ, in the **Link name** field.
2. In the **LAN destination address (hex)** field, enter the value for the DataJoiner workstation. Issue the command `lscfg -1 tok0 -v` on the DataJoiner machine to see the value. For example, 10005AB14931.
3. In the **Partner network ID** field, enter the value for network name in the SNA Control Point Profile. For example, USIBMST.
4. In the **Partner node name** field, enter the value for the CPname in the SNA Control Point profile. For example, DJCP.
5. Select the **Define Partner LUs** push button to proceed to the next step.

**Partner LU Profile:** To create a partner LU profile:

1. On the Connection to a Peer Node dialog, you selected the Define Partner LUs push button. For the Network ID, use the partner network ID used in the Connection to Peer dialog. For example, USIBMST.
2. In the **LU name** field, enter the value for Local LU Name in the SNA LU 6.2 Local LU profile on the DataJoiner machine. For example, STB8004I.
3. Enter an alias value. The alias is an alternative name that transition programs can use when referring to this partner LU. For example, STB8004I.
4. Make sure that the **Partner LU is Dependent** check box is not selected.

**CPI Communications Side Information Profile:** This profile is used to specify the initialization information that the CPI-C subsystem needs to establish a conversation with the TP.

To create a CPI Communications Side Profile:

1. On the Communications Manager List panel, select **SNA features** and click the **Configure** push button. The Features panel displays.
2. Select **CPI communication side information** and click **Create**.
3. Specify a descriptive name for the DataJoiner instance that this client will access, such as LINKDJ, in the **Symbolic destination name** field. This value must match the value that is used later in “Step 2: Catalog the DataJoiner Database” on page 203 .
4. Select the **Alias** push button, and then from the drop down list select the Partner LU Alias you defined in step 3 of “Partner LU Profile”. For example, STB8004I.
5. In the **TP name** field, select the Transaction program name that is specified in the SNA TPN profile on the DataJoiner workstation as discussed in step 2 on page 194. For example, zzservertp.
6. Select the **None** radio button under **Security type**. Selecting **None** does not mean that you will have no security; you can specify a security type later in the node directory.
7. In the **Mode name** field, specify a value that is defined on the DataJoiner workstation, such as IBMRDB.



**Mode Definition Profile:** Use this profile to select an IBM-defined mode or add a new mode definition that is appropriate for your applications.

To create a CPI Communication Side Information Profile:

1. On the Communications Manager Profile List panel, select **SNA features** and click the **Configure** push button. The Features panel opens.
2. Select **Modes** and click **Create**.
3. In the **Mode name** field, enter the Mode Name value that is used on the CPI Communications Side Information Profile. For example, IBMRDB.
4. Enter 30 in the **Mode Session limit** field.
5. Enter 15 in the **Minimum contentions winners** field.
6. Select the **Maximum RU size** radio button and enter 4096.

Refer to online help for information about these values.

## Step 2: Catalog the DataJoiner Database

As the user with sysadm authority on the client machine, enter the following commands to catalog the DataJoiner database on the client:

```
catalog appc node djwkst remote LINKDJ security program
catalog database djdb22 as djdb22 at node djwkst authentication server
```

where:

- *djwkst* is the node entry name. This value correlates databases that are cataloged in the system database directory with an entry in the node directory. Try to use a value that identifies the DataJoiner workstation.
- LINKDJ is the symbolic destination name and must match the value that is specified in the CM/2 CPI Communications Side Information Profile. See “CPI Communications Side Information Profile” on page 202 for details.
- *djdb22* is both the alias and database name for the DataJoiner database in this example. The value for the database name here must match the alias name for the DataJoiner database on the DataJoiner workstation.

Security and authentication values in the commands above indicate that user names and passwords will be sent to the DataJoiner server.

## Step 3: Connect to DataJoiner

For OS/2 clients, the user name and password can be managed by User Profile Management (UPM). Refer to the *DB2 Installing and Using OS/2 Clients V2, S2OH-4782*, for more information about UPM.

Alternatively, you can specify the user name and password item on the CONNECT statement. Use one of these command formats to connect to DataJoiner:

connect to *database*  
connect to *database* user *user* using *password*

where:

- *database* is the alias name of the DataJoiner database. The example uses djdb22.
- *user* is a valid user name on the DataJoiner workstation.
- *password* is the password for the user. Note that OS/2 folds passwords to uppercase. When you add the password to the user name on an AIX workstation, specify it in uppercase.

#### **Step 4: Bind the Database Utilities**

After connecting to the database, you must bind the utilities. The bind is automatic if the user is a database administrator. To explicitly bind the utilities, enter the following commands:

```
bind path\@db2ubind.lst messages bind.msg grant public
bind path\@db2cli.lst messages clibind grant public
```

---

## Chapter 20. Controlling Communication Support Initialization

The DataJoiner administrator uses the DB2COMM environment variable to control which communication protocols are enabled when a **db2start** command is issued. You can set and change DB2COMM any time, and the new value is effective on the next **db2start** command.

Set DB2COMM by using the techniques that are described in “Step 2: Set Environment Variables” on page 57. You can use any combination of the following keywords, separated by commas:

|                |                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TCPIP</b>   | Initializes TCP/IP support.                                                                                                                         |
| <b>APPC</b>    | Initializes APPC support.                                                                                                                           |
| <b>IPX/SPX</b> | Initializes IPX/SPX support.                                                                                                                        |
| <b>NONE</b>    | Prevents the initialization of communication support. This keyword takes precedence; if NONE is specified all other specified keywords are ignored. |

For example, if your server is configured to start TCP/IP, the variable would be set to DB2COMM=TCP/IP. To enable APPC and TCP/IP, set the environment variable to DB2COMM=APPC, TCP/IP.

Similarly, if you set DB2COMM to APPC, set the TPNAME variable in the database configuration file; if you set DB2COMM to TCP/IP, set the SVCENAME variable in the database configuration file.

If DB2COMM is undefined or set to null, no remote communication protocol support is started at the server.



---

## Chapter 21. Configuring the DataJoiner DRDA Application Server and Clients

This chapter describes how to configure DataJoiner for AIX as a DRDA Application Server (DRDA AS) and points you to the relevant resource for configuring clients as DRDA Application Requesters (DRDA ARs). A sample on how to configure a DB2 for MVS DRDA AR to DataJoiner is provided in “DB2 for MVS DRDA AR Sample” on page 213 .

---

### APPC Support

DataJoiner DRDA Application Server requires APPC support, which DataJoiner provides through the SNA Clients option (djk\_02\_01\_01.cs.sna) during DataJoiner installation. To use APPC, SNA Server/6000 Version 2 must be installed and configured on the DataJoiner server workstation.

---

### Configuring DataJoiner as a DRDA AS

To configure DataJoiner as a DRDA AS, complete the following steps:

1. Configure SNA Server/6000.
2. Update the database manager configuration.
3. Stop all connections.
4. Restart the instance.

Details for each step follow.

#### Step 1: Configure SNA Server/6000

As root user:

1. Create AIX SNA Server/6000 profiles to define the SNA environment if they do not already exist. These are:
  - Node profile (see Figure 17 on page 91)
  - Control point profile (see Figure 12 on page 82)
  - DLC profile (see Figure 10 on page 80)
  -

**Optional:** Link station profile (see Figure 11 on page 81). This step is required if the link to DB2 for MVS is started from the DataJoiner machine, and may be helpful when no PATH to DataJoiner has been specified in the VTAM Switched Major

Node Definition. If, however, the PATH is completed, then the link can be initiated from DB2 for MVS and this profile is no longer necessary.

- Mode profile (see Figure 15 on page 87)
  - Local LU profile (see Figure 13 on page 84) is required unless the control point name is the independent LU name that is defined in VTAM.
2. Create an LU 6.2 TPN profile. This TPN should match your database manager configuration file's tpname, which is set in "Step 2: Update the Database Manager Configuration" on page 212. Complete the following steps:
- a. From the LU 6.2 menu, select **LU 6.2 Transaction Program Name (TPN)** and then select **Add a Profile**. The **Change/Show LU 6.2 TPN Profile** window opens.

```

Change/Show LU 6.2 TPN Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...1] [Entry Fields]
New profile name []
Transaction program name (TPN) [zzservertp]
Transaction program name (TPN) is in hexadecimal? no +
PIP data? no +
 If yes, Subfields (0-99) [0] #
Use Command Line Parameters? no +
Command Line Parameters []
Conversation type basic +
Sync level none/confirm +
Resource security level none +
 If access, Resource Security Access List Prof. []
Full path to TP executable <dj/sqllib/bin/db2acntp]
Multiple instances supported? yes +
User ID [221] #
[MORE...9]

F1=Help F2=Refresh F3=Cancel F4=List
F5=Reset F6=Command F7=Edit F8=Image
F9=Shell F10=Exit Enter=Do

```

Figure 45. TPN Profile Screen

- b. Enter a profile name in the **New profile name** field. Consider using your actual TPN.
- c. Enter a transaction program name in the **Transaction program name (TPN)** field. You will need to enter this same name in the system management

configuration file as the Transaction Program Name (TPNAME), which is described in “Step 2: Update the Database Manager Configuration” on page 212

- d. Set the **Transaction program name is in HEXADECIMAL?** field to no.
  - e. Set the **Conversation type** field to basic.
  - f. Set the **Full PATH to TP executable** field to `inst/sql1lib/bin/db2acntp`, where *inst* is the home directory for the instance owner. For example, `/home/instdj`.
  - g. Set the **Multiple instances supported?** field to yes.
  - h. Set the **User ID** field to the instance owner’s user ID number (not the user name). You can obtain this value by entering the `id` command while logged on as the instance owner. You would get a result such as:  
`uid=221`
  - i. Use default values in the remaining fields.
  - j. Press Enter to save the profile.
3. Create an interrupt manager TPN profile by completing the following steps.

**Note:** This step is required only for DB2 CAE V1 clients. For more information, see the *DDCS for AIX Installation and Configuration Guide* (S20H-4794).

- a. From the LU 6.2 Transaction Program Name menu, select **Add a Profile**. The **Change/Show LU 6.2 TPN Profile** window opens

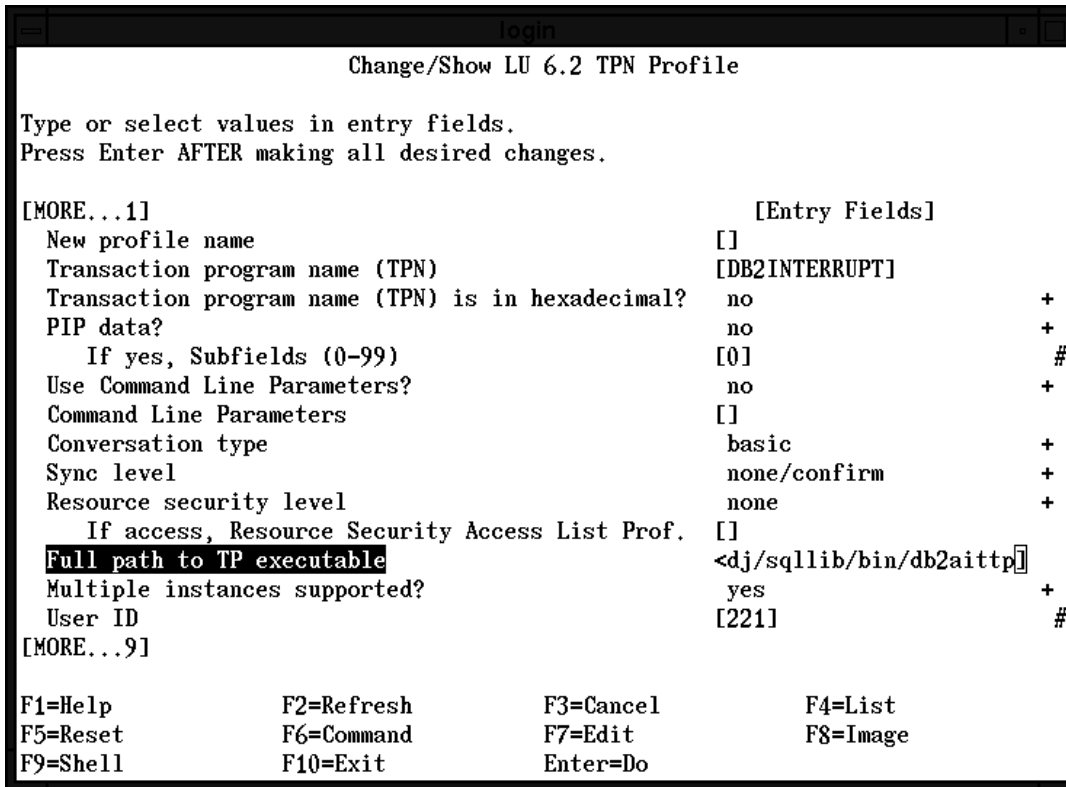


Figure 46. TPN Profile for AIX Client

- b. Enter a profile name in the **New profile name** field. Consider using a name similar to the previous communication manager TPN name, such as `zzserverint`.
- c. Set the **Transaction program name (TPN)** field to `DB2INTERRUPT`.
- d. Set the **Transaction program name (TPN) is in HEXADEDECIMAL?** field to `no`.
- e. Set the **Full path to TP executable** field to `inst/sqllib/bin/db2aittp`, where `inst` is the home directory for the instance owner. For example, `/home/instdj`.
- f. Set the **Multiple instances supported?** field to `yes`.
- g. Set the **User ID** field to the instance owner's user name number (not the user name). You can obtain this value by entering the `id` command while logged on as the instance owner. You would get a result such as:
 

```
uid=221
```
- h. Use default values in the remaining fields.
- i. Press Enter to save the profile.
- j. Press F3 to return to the **LU 6.2 Transaction Program Name** menu.



4. Create a Partner LU Profile for each DRDA requester. For the Conversation Security Level, specify `already_verified` to support APPC security type SAME.

```

X: aixterm
Change/Show LU 6.2 Partner LU Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Current profile name [Entry Fields]
 dpdjec121u
New profile name []
Fully qualified partner LU name [USIBMST.STIIDB2A]
Partner LU alias [EC12DB23]
Parallel sessions supported? yes +
Session security supported? no +
Conversation security level already_verified +
Comments []

F1=Help F2=Refresh F3=Cancel F4=List
F5=Reset F6=Command F7=Edit F8=Image
F9=Shell F10=Exit Enter=Do

```

Figure 47. Partner LU Profile, Conversation Security Level

5. Add the primary group name of the DataJoiner instance to the trusted group name for SNA. You can use the following procedure:
  - a. Determine the group name of the instance. It was created by following the procedure on “Step 1: Create an Instance of the Product” on page 55. You can also determine the group by using the `ls -l` command on the object `INSTHOME/sqllib/adm/db2sysc`, since it belongs to the same group as the instance.
  - b. Use **smit** to update the SNA configuration.

Choose the screens in the following order to update the SNA System Defaults Profile:

    - SNA System Defaults
    - Change/Show a Profile

Add the instance group names to the list in the Trusted Group Names field.
6. As root, start SNA Server/6000 and the link station if it is configured. Use smitty, or enter the following commands:

```

startsrc -s sna
startsrc -s l -p 'link_station_profile_name' &

```

## Step 2: Update the Database Manager Configuration

As the person with sysadm authority for the instance, use the following command to update the TPNAME field in the database manager configuration file. This name must match the LU 6.2 TPN name being used by the remote APPC client.

```
db2 update database manager configuration using tpname ZZSERVERTP
```

## Step 3: Set DB2COMM for APPC

To enable APPC support when the **db2start** command is issued, you must first set the instance owner's DB2COMM environment variable to a value of APPC. Use the following command:

```
export DB2COMM=APPC
```

Also, update any existing DB2COMM variable settings with APPC; for example, if the existing value is:

```
DB2COMM=TCPIP
```

change it to:

```
DB2COMM=TCPIP,APPC
```

See the *Database 2 for AIX Installation and Operation Guide*, (20H4757) for more information.

## Step 4: Stop All Connections

Stop all connections to the instance with either of the following commands:

- **db2 force application all**
- **db2stop**

## Step 5: Restart the Instance

To restart the instance, enter the **db2start** command. Several listen processes start and wait to be contacted by remote clients using a transaction program name defined in a TPN profile.

For further information on configuring DataJoiner as a DRDA AS, see *DB2 for MVS Connections with AIX and OS/2*, (SG24-4558-00) and "Appendix A. DRDA Application Server" on page 307.

---

## Configuring Clients as DRDA ARs

To configure clients as DRDA ARs, refer to "Setting Up the Application Requester" for your specific client and platform in the *Distributed Relational Database Architecture Connectivity Guide*, (SC26-4783-03).

---

### DB2 for MVS DRDA AR Sample

This section gives an example for configuring a DB2 for MVS DRDA AR to DataJoiner. Figure 48 on page 215 describes the configuration. Figure 49 on page 216 is provided for your use in defining your own configuration. The following steps take you through the process. For more information, refer to the *DRDA Connectivity Guide*.

1. Perform the steps that are described in "Configuring DataJoiner as a DRDA AS" on page 207 . It is assumed the VTAM definitions supporting the AIX workstation have already been made.
2. It is also assumed the following has been completed on the DB2 for MVS system:
  - DDF is configured.
  - The DB2 for MVS system has already been defined to VTAM, including the NETID for DB2 and the VTAM APPL definition.
3. Update the DB2 for MVS communications database (CDB) to define each remote DataJoiner DRDA AS. The CDB is described in detail in the *Database 2 for MVS/ESA SQL Reference*, SC26-3270.

- a. Update SYSIBM.SYSLOCATIONS with each DataJoiner database, for example:

```
insert into SYSIBM.SYSLOCATIONS (location,loctype,linkname,
linkattr) values('dbdb1',' ','STB004I','ZZSERVERTP')
```

**location**

The remote database name or alias.

**loctype**

Leave blank.

**linkname**

The workstation's independent LU as it is defined to VTAM, which must match the LUNAME in SYSLUNAMES and SYSUSERNAMES. Note that it can be the control point name or the LU name that is defined in SNA on the DataJoiner workstation.

**linkattr**

The TPN profile, which must match the DataJoiner tpname in SNA Server, and in the DataJoiner database manager configuration file.

- b. Update SYSIBM.SYSLUNAMES for each DataJoiner instance. The minimum entry that is required is:

```
insert into SYSIBM.SYSLUNAMES(luname,sysmodename,usersecurity,
encryptpswds,modeselect,username) values('STB8004I',' ','A',
'N','N',' ')
```

The columns of particular interest in an AR configuration are these:

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>luname</b>       | The workstation's independent LU as it is defined to VTAM, which must match the LINKNAME column of SYSLOCATIONS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>usersecurity</b> | The security acceptance option for <i>inbound</i> requests from the remote LU. <ul style="list-style-type: none"><li>• 'A' – verification already performed (AUTHID required, password optional)</li><li>• 'C' – all conversations need verification (both AUTHID and password are required)</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>usernames</b>    | The type of translation performed for IDs <ul style="list-style-type: none"><li>• '' – No translation, Security Same assumed. Only the AUTHID is sent with an already verified indicator. Use when the DataJoiner database is cataloged with AUTHENTICATION CLIENT.</li><li>• 'O' – Outbound translation, Security Program assumed. Use when theDataJoiner database is cataloged with AUTHENTICATION SERVER or DCS. The AUTHID and Password that are located in SYSUSERNAMES are sent, so ensure that SYSUSERNAMES contains an entry for each MVS user ID that will connect to the DataJoiner database. DB2 for MVS uses SYSUSERNAMES because the MVS logon and password cannot be determined when the CONNECT statement is issued.</li><li>• 'B' – Translation occurs both inbound and outbound.</li><li>• 'I' – Inbound translation. This value is not used in a DB2 for MVS AR configuration.</li></ul> |

- c. Update SYSIBM.SYSUSERNAMES if USERNAMES has been specified as 'O' or 'B' in SYSLUNAMES as described above. In Figure 48 on page 215, no translation is used.

All passwords in this table are not encrypted because DB2 for MVS can only accept encrypted passwords from another DB2 for MVS AS. The administrator should thus consider doing two things:

- 1) Since passwords are not encrypted, restrict access to this table.
- 2) Create a view that will allow each user to update their own password through the view.

- d. Update SYSIBM.SYSMODESELECT with the value 'Y' to associate a mode with an outgoing SQL request. In Figure 48 on page 215, because no association was made, 'N' was specified.

4. To execute applications from MVS, they must first be bound to the DataJoiner database. In Figure 48 on page 215, DJDB1 is the database.

Refer to *DB2 for MVS Connections with AIX and OS/2*, (SG24-4558-00) for information about binding applications to DRDA servers.

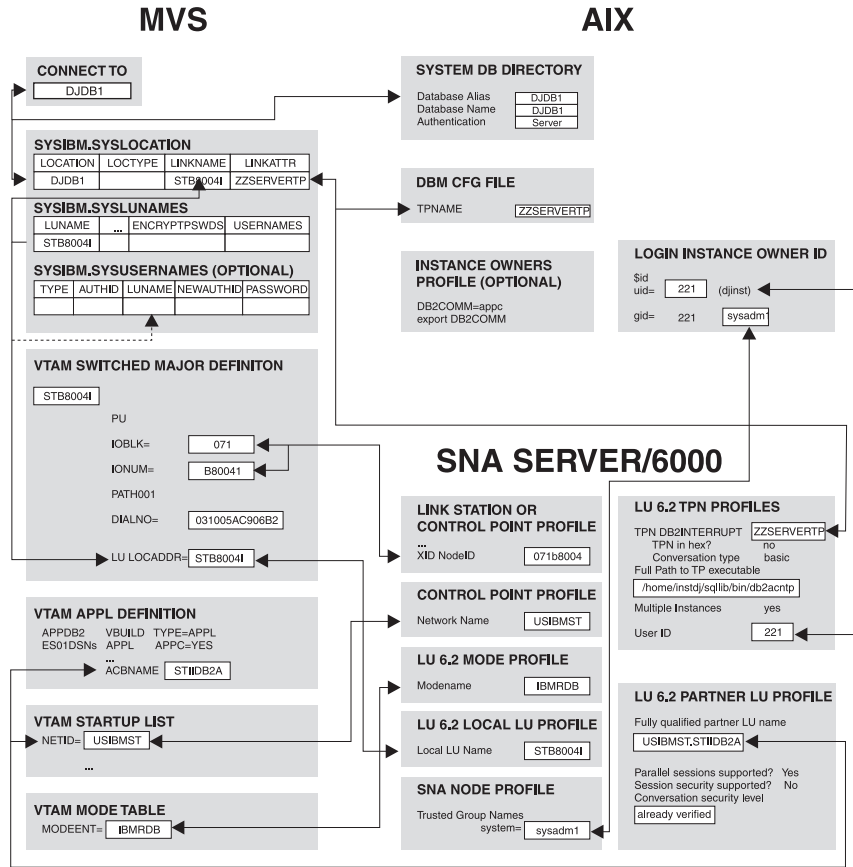


Figure 48. DB2 for MVS DRDA AR Configuration

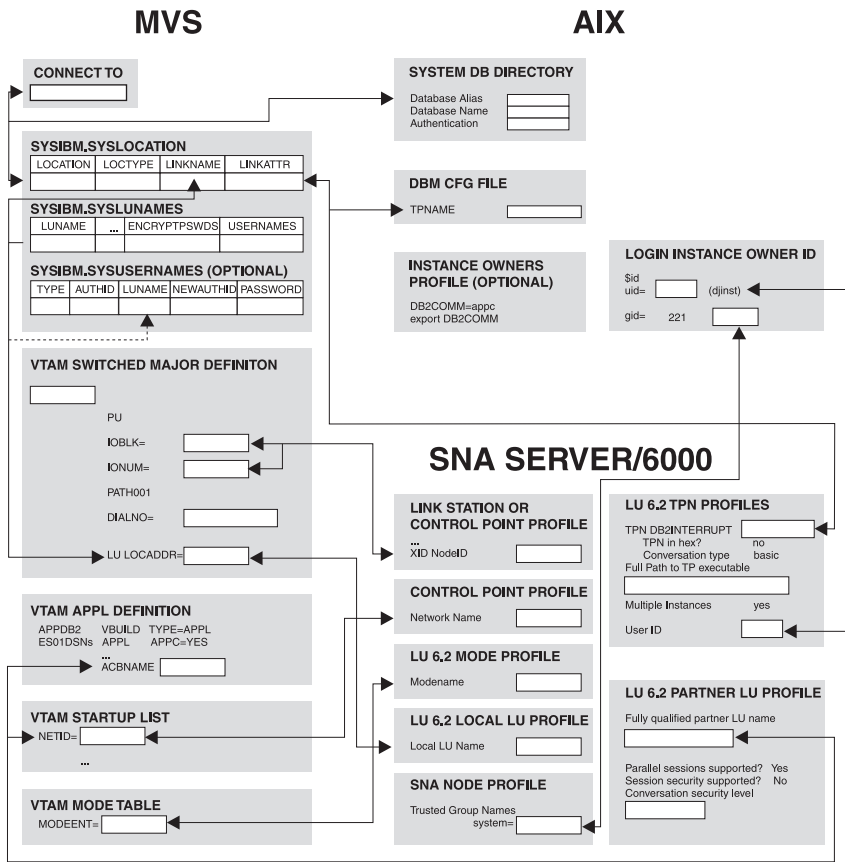


Figure 49. DB2 for MVS DRDA AR Configuration Template

---

## Part 5. Maintaining DataJoiner





---

## Chapter 22. Maintaining the DataJoiner Environment

This chapter contains information on possible considerations when maintaining a DataJoiner environment.

---

### DB2 and DDCS Coexistence with DataJoiner

DataJoiner instances can coexist on the same workstation as DB2 for AIX, DB2/6000, DDCS for AIX, and DDCS/6000 instances. However, symbolic links that are created by the DB2 and DDCS **db2ln** command affect the DataJoiner application programming environment on the same workstation where DataJoiner is installed.

You can use the **db2ln** command to create symbolic links between DB2 and DDCS product directories and `/usr/include` and `/usr/lib` for certain header files, such as `.h`, `.cbl`, and `.etc`, and for libraries, such as `libdb2.a`. This command is optional; you can use it to compile and link-edit applications without specifying options on the compile and link-edit commands; for example, `-I` (uppercase i), `-L`, and `-l` (lowercase L) on the **cc** command, and `-L` and `-l` (lowercase L) on the **ld** command.

If DDCS or DB2 already exists on the same system where DataJoiner is being installed, these symbolic links might already exist. DataJoiner uses some of the same header file and library names as DB2 and DDCS. However, the files are not compatible between DataJoiner and the other products. Therefore, when compiling a DataJoiner application, you must use the `-I` (uppercase i) option to specify the full path for header files. To identify the correct libraries you can either:

- Use the `-L` compiler/link-editor options to specify the full path for libraries as in:  
`-L/usr/lpp/djx_02_01_01/lib`

However, this option leaves your applications dependent on that particular version and release of DataJoiner. If you upgrade DataJoiner to a new version or release, the directory of the new version or release will be different from the old version or release, and you must relink-edit all of your applications.

- Specify the full path for the libraries in your `LIBPATH` environment variable. `LIBPATH` is used by DataJoiner to locate the required libraries. The search order for libraries is:
  1. Directories in `LIBPATH` (if any)
  2. Directories specified using the compile and link-edit time option `-L`
  3. Default libraries (`/lib` and `/usr/lib`)

This option does not make your application dependent on a particular version or release of DataJoiner, but it might affect other applications because many programs can use `LIBPATH` to locate libraries.

---

## Password Considerations for OS/2 Clients

If you plan to have a network that includes OS/2 database clients, specify logins in lowercase and passwords in uppercase on the UNIX platform.

---

## Maintaining DataJoiner

To ensure that your version of DataJoiner is kept current, you can install fixes as they are provided, and delete uncommitted options that are not necessary for your system.

## Installing DataJoiner Program Temporary Fixes (PTFs)

To install DataJoiner PTFs:

1. Stop all DataJoiner instances.
2. Apply the PTF by using `smitty` or `installp`.
3. Read the file, `/usr/lpp/djx_02_01_01/lpp.README`, and pay particular attention to section 1.0 "User Impact."

The first two steps can be performed in either order, but they *must* be performed before doing the remaining steps.

4. From the root login, run `/usr/lpp/djx_02_01_01_0000/instance/db2instupdt` for each DataJoiner instance you have. For example:

```
/usr/lpp/djx_02_01_01_0000/instance/db2instupdt djinst1
```

5. Restart each DataJoiner instance.
6. Make any necessary changes as described in section 1.0 of the README file.

## Deleting Options on AIX 3.2

In AIX 3.2, the procedure you follow to remove an option depends on whether the option is in the applied or committed state.

- If the option is in the applied state, use either the Remove Applied Software Product menu in `smitty` or the **installp** command with the `reject` flag. For example:

```
installp -r djx_02_01_01.dd
```

You cannot remove an option if it is a prerequisite for other options that are installed on the system unless you also remove the other options at the same time. To remove all the DataJoiner options for a given version, enter the **installp** command as follows:

```
installp -r djx_02_01_01
```

This command:

- Updates the SWVPD database.
- Deletes all the installed files and directories.
- Undoes all pre-installation and post-installation processing.

| The /usr/lpp/djx\_02\_01\_01 directory is not removed when all the DataJoiner options  
| are deleted. It is, however, emptied.

- If the option is in the committed state, the only way to remove it is to reinstall it using the force option and then to reject it. To perform this procedure:
  1. Login as root.
  2. Execute **installp** to do a force reinstallation that will leave the option in applied state.
  3. Execute **installp** with the reject flag, as described earlier in this section.

## Deleting Options on AIX 4.1

On AIX 4.1, options that are successfully installed are always in the committed state. To remove options, use the deinstall (-u) switch. For example:

| `installp -u djx_02_01_01`

This command removes all product options that match the djx\_02\_01\_01 prefix from the system. You can use the deinstall switch to remove product options that are in the broken state.



---

## Chapter 23. Maintaining Instances

This chapter covers issues that pertain to instances that are already installed.

---

### Starting and Stopping a DataJoiner Instance

You can use DataJoiner or operating system commands to control instances. Use the operating system **kill** command only as a last resort.

#### Using DataJoiner Commands

You must start a DataJoiner instance before you can connect to a database, precompile an application, or bind a package to a database. The commands to start and stop an instance are, respectively:

**db2start**

**db2stop**

When you want to terminate an instance, use the **db2stop** command, and, if necessary, the **force** command to terminate all active applications that connect to the database.

See the *DB2 Command and API Reference* for more information about DataJoiner commands.

#### Using the kill Command

Use the operating system **kill** command to terminate a DataJoiner instance only if the DataJoiner commands do not work. Using the operating system **kill** command abruptly ends DataJoiner processes without the appropriate termination and clean-up processing. If the **kill** command is used, use the **ipcs** and **ipcrm** commands to identify and remove message queues and some memory sets.

---

### When to Stop and Restart Instances

Certain changes to your system require you to stop and restart (recycle) a given DataJoiner instance before the changes become effective for that instance. These changes are:

- When you modify any environment variable that is used by the instance, such as those listed in “Step 2: Set Environment Variables” on page 57. The environment variables a DataJoiner instance can use are:

DJX\_NR\_CONFIG (for the Classic Connect interface)

DJXCOMM

INFORMIXDIR (for Informix V5 and V7 data sources)

SQLRM (for Informix V5 data sources)

INFORMIXSERVER (for Informix V7 data sources)  
ORACLE\_HOME (for Oracle data sources)  
TNS\_ADMIN (for Oracle data sources)  
SYBASE (for Sybase data sources)  
PATH  
LANG  
HOME

Some of these variables are specific to a data source. If you do not have the client libraries for a data source link-edited to DataJoiner, the environment variables that are associated with the data source do not affect DataJoiner.

The instance might reference additional environment variables if you set up a Generic data access module as described in “Chapter 14. Accessing Data Sources Using a Generic Data Access Module” on page 165.

- When you modify any of the tuning parameters and the configuration parameters in the instance’s database manager configuration file. The database manager configuration file is discussed in the *DB2 Administration Guide*.
- When you modify any of the configuration parameters in the Classic Connect interface configuration file *except* for the following:
  - AF TRACE LEVEL
  - AF WAIT TIME
  - EXTERNAL TASK WAIT TIME
  - FETCH BUFFER SIZE
  - TASK CONNECT WAIT TIME

Though these parameters do not require that you recycle the instance, users currently accessing Classic Connect must issue the **connect reset** command and reconnect to the database before seeing the change.

The following situations require you to stop and restart all DataJoiner instances:

- When you run the `djxlink.makefile`, and instances already exist. `djxlink` is discussed in “Step 3: Link DataJoiner to Data Source Client Software” on page 47.
- When DataJoiner service is applied.
- When you create a data access module.
- When you update an instance by using the **db2iupdt** command.

---

## Updating Instances

Existing instances are designed to be as independent as possible from the effects of subsequent installation and removal of product options. In general, the installation or removal of product options does not require any changes to existing instances. In most cases, existing instances will automatically inherit or lose access to the function of the

option that is being installed or removed. For example, if an option such as a message catalog is installed or removed, the new function is automatically inherited or lost by all existing instances.

However, when the following options:

- djx\_02\_01\_01.db2.rte (DataJoiner Executables)
- djx\_02\_01\_01.cs.rte (DataJoiner Communication Support - Base with TCP/IP)
- djx\_02\_01\_01.dd (Database Director)

are installed or removed, existing instances do not automatically inherit the new system configuration parameters or gain access to all the additional function.

To access new configuration and function, existing instances must be updated using the **db2iupdt** command in `/usr/lpp/djx_02_01_01/instance`.

The db2iupdt script can be invoked as follows:

```
▶▶—db2iupdt—instanceName—-u—UdfName—▶▶
```

where

**instanceName**

is the name of the instance to be updated

**-u** is an optional parameter that specifies the AIX user and group under which Fenced UDFs and DARIs will execute.

**Restriction:**

*UdfName* may not be root or bin.

If *UdfName* is unspecified when you update an instance, one of two actions will occur:

- The user and group names for UDFs will remain unchanged if they were already defined.
- The user and group names for UDFs default to nobody:nobody; this means that there are no file access permissions defined for UDFs or DARIs.

Running the db2iupdt script updates the specified instance by performing the following processing:

1. Replace the files in `INSTHOME/sqllib/adm` with the files in `/usr/lpp/djx_02_01_01/adm`.
2. If the node type is changed, then a new database manager configuration file is created by merging relevant values from the existing database manager configuration file with the default database manager configuration file for the new node type. If a new database manager configuration file is created, the old file is backed up to `INSTHOME/sqllib/db2system.bak`.

---

## Starting a DataJoiner Instance at System Boot

To start a DataJoiner instance when the AIX system is booted:

1. Login as root.
2. Use an editor to create a file named `/etc/rc.djx`. Enter the following line for each instance that is to be started at system boot:

```
su - instname "-c db2start > /dev/console 2>&1"
```

where *instname* represents the instance name.

3. Set the permissions on the `/etc/rc.djx` file to 744 by entering:  

```
chmod 744 /etc/rc.djx
```
4. Add an entry, `/etc/rc.djx`, to the `/etc/rc.djx` file.

---

## Multiple Instances

DataJoiner allows you to run multiple instances on the same operating system. Each instance must have a unique name. For a local client to use a particular instance, you must set an environment variable to the name of the instance, and you must have certain directories in your path. For more information, see “Step 2: Set Environment Variables” on page 57.

Instances are virtual copies of the multidatabase server that are designed to isolate one environment from another. An installation can isolate its production environment from any effects of its test environment by defining a distinct instance for each environment. Use multiple instances when load requirements necessitate multiple DataJoiner servers running on the same machine and sharing the same set of data sources. In such cases, you can configure each instance differently with respect to allocation of system resources. Each such DataJoiner instance is isolated from the failure of other DataJoiner instances.

An instance has its own owner, system administrator, databases, and users. Only the root administrator can create an instance, and only the root administrator or the instance owner can remove an instance. The SYSADM authority is controlled at the instance level; other privileges are controlled at the database and object level. Refer to the *DataJoiner Administration Supplement* for more information. The security mechanisms of each data source control data source access.

There are several of reasons that you might want to create multiple instances of DataJoiner. Some possibilities are:

- To isolate a development and testing environment from production
- To tune the database manager configuration file for a particular database environment
- To restrict sysadm access to certain databases



You can create additional instances by following the instructions in “Step 1: Create an Instance of the Product” on page 55. There must be a unique correspondence between an instance and an instance owner. Also, each instance owner must have a unique home directory.

To separate sysadm authority between instances, ensure that each instance owner belongs to a different primary group before creating instances. Otherwise a single user could have sysadm authority over multiple instances. See Figure 50.

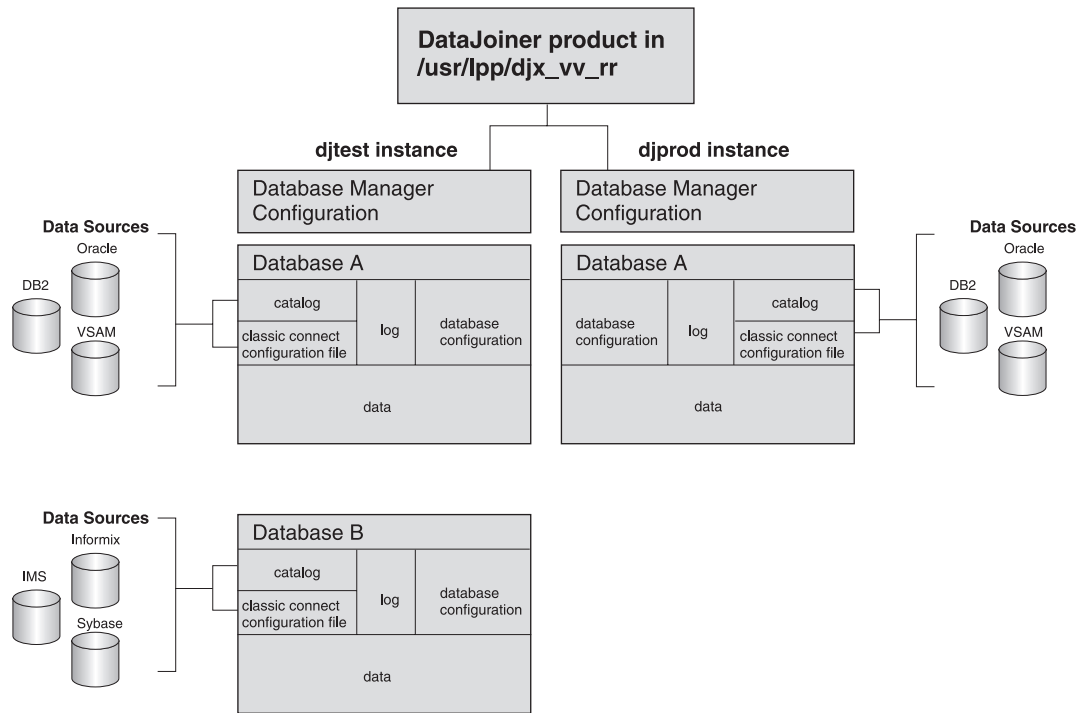


Figure 50. DataJoiner Instances

This figure shows a test instance and a production instance. In this configuration:

- The two instances represent production and test environments that are based on the same release of DataJoiner (02\_01\_01). They could, however, represent test environments for two different releases of DataJoiner. For example, they could represent a migration from djx\_02\_01 to djx\_02\_02 (that is, release 1 to release 2).
- sysadm access is controlled by membership in the primary group that is associated with the djtest and djprod user IDs.
- Database manager configuration affects all databases in a given instance. Service names for remote TCP/IP clients, tnames for remote APPC clients, and specific tuning parameters for that instance are contained in the Database manager

configuration. This configuration separates users who are accessing DataJoiner as remote clients, using the test instance, from users who are using the production instance.

- The test instance has both a database A and B. Database A is for an application in production so it is also in the production instance. This configuration provides a migration path from a test environment to a production environment.
- Each instance has its own log.
- Each instance has its own catalog and Classic Connect configuration file for configuring to data sources that are accessed through DataJoiner. The test and production instances can reference the same data sources or different data sources.
- The test and production instances are separated: djtest can be stopped and started, databases updated, and applications changed without fear of affecting production users.

---

## Removing Instances

A DataJoiner instance that was created following the procedure that is described in “Step 1: Create an Instance of the Product” on page 55 can be removed as follows:

1. End all applications that are currently using the instance.
2. Stop the database manager and the command line processor.  
Issue the **db2stop** and **db2 terminate** commands.
3. Backup files in the INSTHOME/sql1lib directory if needed. Backup candidates include your db2system file and DARI applications kept in INSTHOME/sql1lib/function, where INSTHOME is the home directory of the instance owner.
4. As root, execute the **db2idrop** command in /usr/lpp/djx\_02\_01\_01/instance as follows:

```
db2idrop instanceName
```

where *instanceName* is the name of the instance being dropped.

The **db2idrop** command removes the instance entry from the list of instances and erases the INSTHOME/sql1lib directory.

5. As root, remove the instance owner's user ID and group if used only for that instance. Do not remove these if you are planning to recreate the instance.

To recreate an instance, see “Step 1: Create an Instance of the Product” on page 55.

---

## Authorizing Access within an Instance

DataJoiner provides optional group level authorization by using the underlying group capabilities of the operating system. Group authorization must be explicitly enabled by the system administrator, who sets the environment variable DB2GROUPS to ON. (DB2GROUPS can be set in db2profile as described in “Step 2: Set Environment Variables” on page 57.)

If this variable is defined, the system enables group support when the instance is started. Follow these steps to enable group authorization:

1. Log in as the instance owner of DataJoiner.
2. Set DB2GROUPS to ON by using whichever method is appropriate for your shell environment.
3. Recycle (stop and restart) the DataJoiner instance.

After these steps, DataJoiner will run with group authorizations enabled. For more information see the *DB2 Administration Guide*.



---

## Part 6. 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,<sup>3</sup> 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 MS SQL Server) is identified (or defined) to DataJoiner to be accessed for replication. When a DataJoiner database is identified as a source or target server, the heterogeneous server must be identified to DataJoiner, specifying the specific Oracle, Sybase, MS 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.

---

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

**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 24. 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 (UDB): The Capture and Apply programs packaged<sup>4</sup> 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
- DataPropagator NonRelational (DPROP NR)
- 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,<sup>5</sup> brings IMS, VSAM, and flat file data into the staging area, making it available for replication to client/server targets.<sup>6</sup> 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.

---

## Replication Solution and Product Environment

Figure 51 on page 235 conceptually shows the relationships between the components of IBM's replication solution.

---

4. The Capture and Apply programs are separately-priced features of DB2 UDB for OS/390. They are packaged with UDB on Windows 9.x and NT, OS/2, and on AIX and other UNIX platforms.

5. 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 for MVS products.

6. See <http://www.software.ibm.com/data/dpropr/> on the Internet for more information about DPROPNR and DataRefresher.



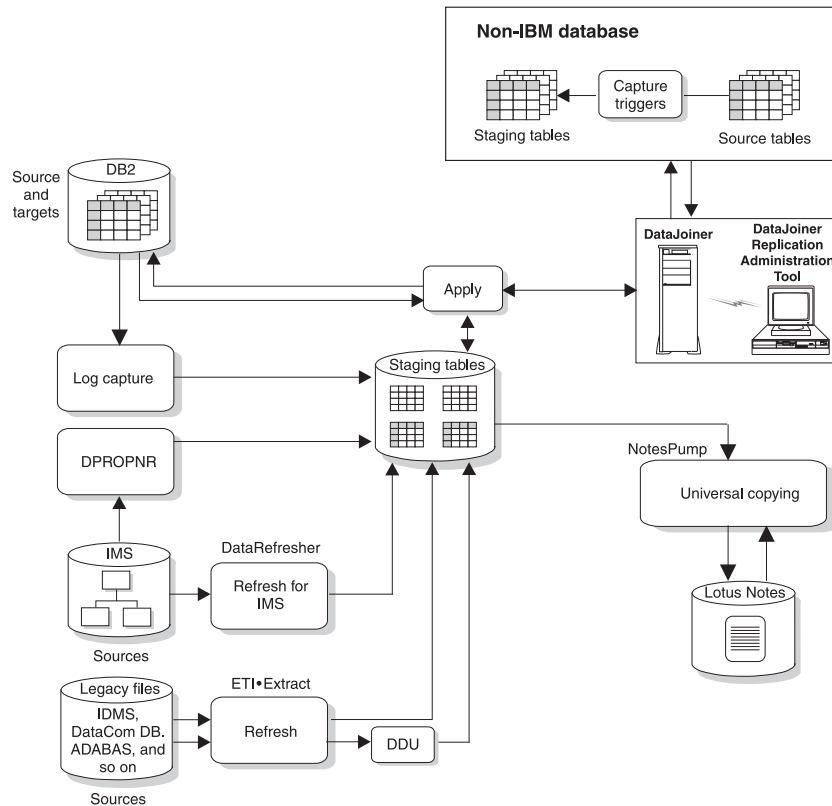


Figure 51. IBM's Data Replication Solution and Product Environment

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 25. 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.

---

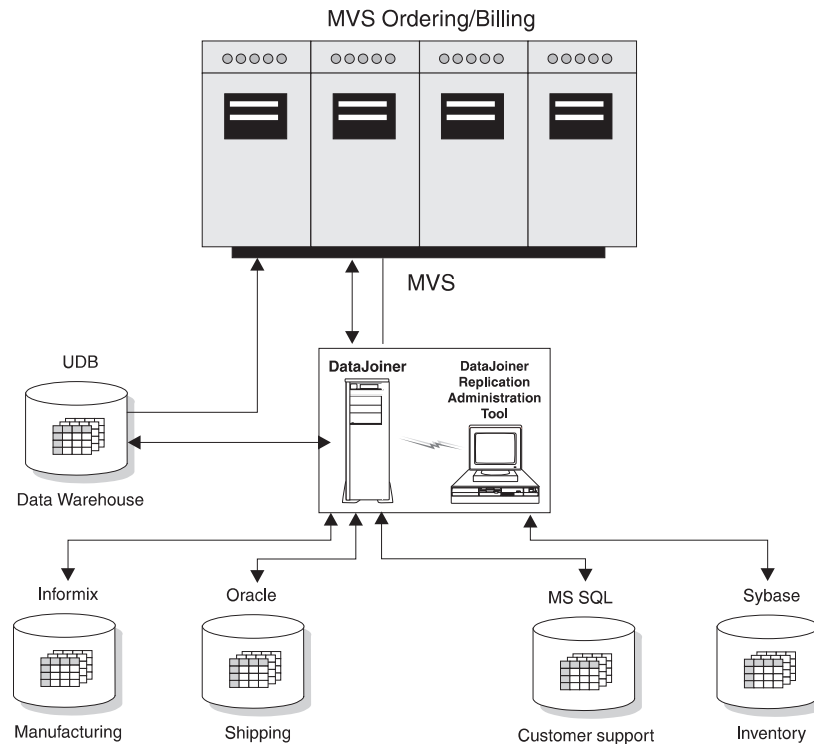
### The Replication Administration Tool

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 replication administration tool, working with DataJoiner, the Capture program, Capture triggers, and the Apply program, replicates relational data from a variety of sources to a variety of targets. The databases that the replication administration tool supports as sources or targets are:

- DB2 UDB V5
- DB2 for MVS V3R1 or V4R1, and DB2 for OS/390 V5R1
- DB2 for common servers V2 and DataJoiner V2
- Oracle V7.0.13
- Informix V7.1 or later
- Sybase V4R6 or later (AIX) and Sybase V11 (Windows NT)
- Sybase SQL Anywhere Version 5.0 (Windows NT)
- Microsoft SQL Server V4.21 or later (AIX) and Microsoft SQL Server V6.0 or later (Windows NT)
- DB2 DPROF 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 52. 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 52. 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 four areas that deal with control tables, sources, subscription sets, and the running or editing of SQL (see Figure 53 on page 240).

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.

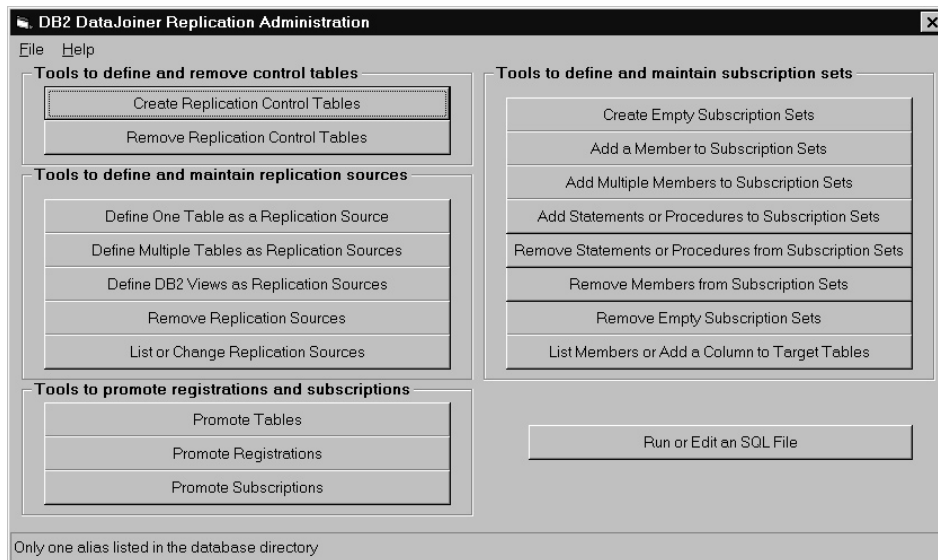


Figure 53. The Replication Administration Tool's Primary Window. The replication administration tool is divided into four areas: control tables, sources, subscription sets and SQL.

## 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 246, "Control Tables Used by the Apply Program" on page 249, and the *IBM DB2 Replication Guide and Reference: V5*.

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 289 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 295 and “Editing the Replication Administration Tool-Generated SQL” on page 297 for more information about generating and editing SQL by using the replication administration tool. See “Running the Replication Administration Tool-Generated SQL” on page 298 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 295 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.

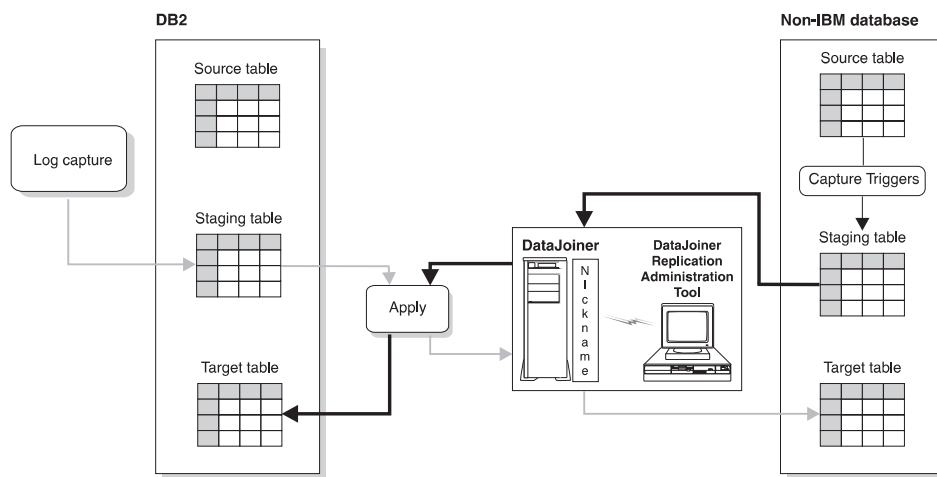


---

## 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 54).



*Figure 54. 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* 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 254 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 257.

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 55 on page 245 shows the relationship between the Capture triggers, the source table, the register control table, and the CCD table.

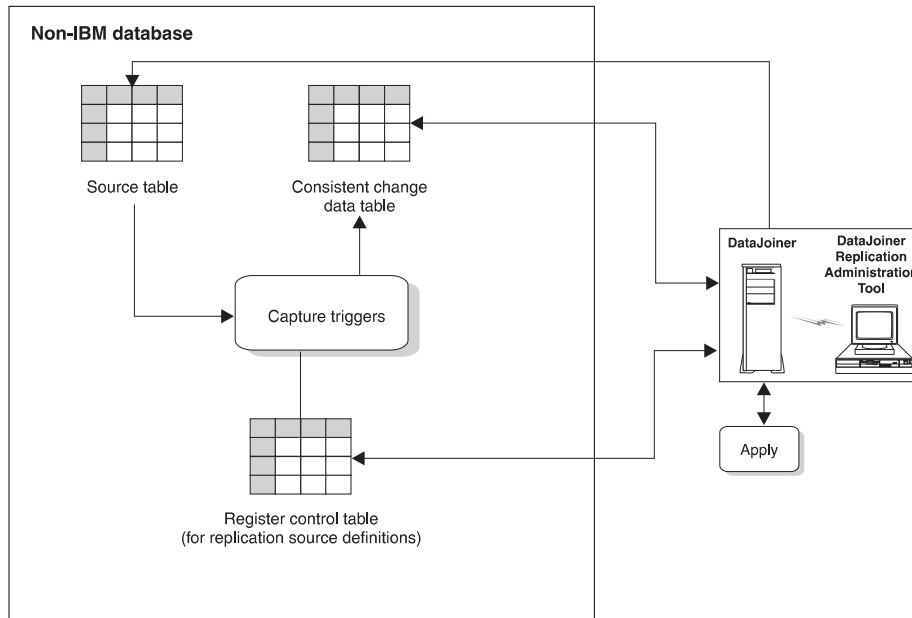


Figure 55. Capture Triggers at the Source Server. The Capture triggers monitor source changes, capture the changed data, and write the changed data to the CCD table.

## 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 257 .

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*.

---

## 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 261 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 249.

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 56 on page 248 shows the Apply program’s relationship with the source tables, the source server control tables, the subscription definition control tables, and the target table.

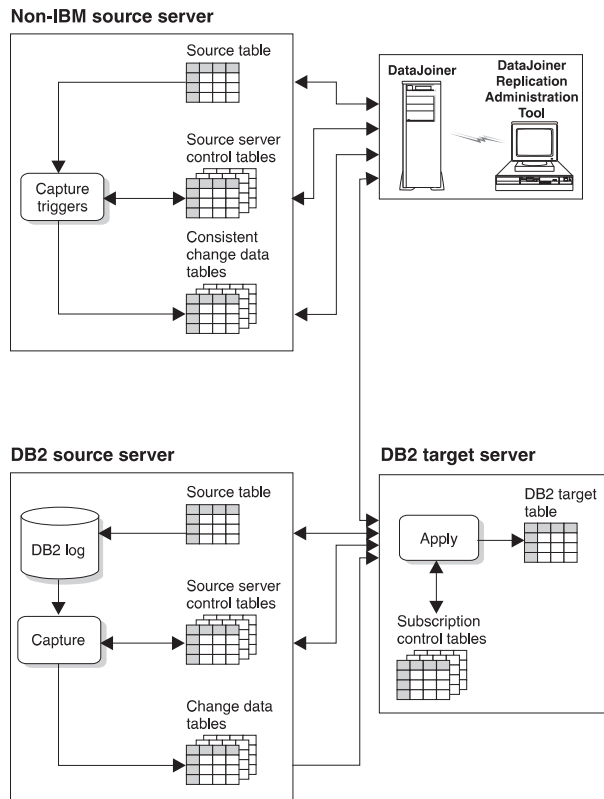


Figure 56. The Apply Program. The Apply program reads the data from the source server tables, manages the source server control tables, manages the subscription control tables, and updates the target table.

## 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 feature<sup>7</sup>, you can have the

7. The Apply program password file is different from replication administration tool's password file.

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 tables<sup>8</sup>) 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.

#### 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.

---

8. 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.

**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*.



## 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 6 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 6. 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  |
| 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  |

## Improving Apply Performance for Sybase or Microsoft SQL Server on AIX

If you are running the Apply program on AIX with a DBLIB connection for either Sybase or Microsoft SQL Server, you can significantly improve your overall performance. To take advantage of this improvement:

1. Create or change the `apply_name.ini` file in the `sql11ib` directory. This file must contain the package names of the Apply program. To find your package names, look in the `applyur.lst` and `applycs.lst` files.

```
ASN6A001+
ASN6B001+
ASN6C001+
ASN6F001+
ASN6I001+
ASN6M001+
ASN6P001
```

*Figure 57. Sample apply\_name.ini File*

2. Create server options for the Apply packet and buffer sizes.

```
create server option apply_packet_size for server type sybase setting 16384;
create server option apply_buffer_size for server type sybase setting 16384;
```

*Figure 58. Sample Server Options for Sybase*

```
create server option apply_packet_size for server type mssqlserver setting 16384;
create server option apply_buffer_size for server type mssqlserver setting 16384;
```

*Figure 59. Sample Server Options for Microsoft SQL Server*

You can set the packet and buffer size to any appropriate value; set them to 16384 initially, and adjust as necessary.

3. Set the following environment variable:

```
DJX_ASYNC_APPLY=TRUE
```

---

## Chapter 26. 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*.

---

### 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* 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* 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* 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*, 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* 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 60 on page 256 shows the columns in a CCD table.

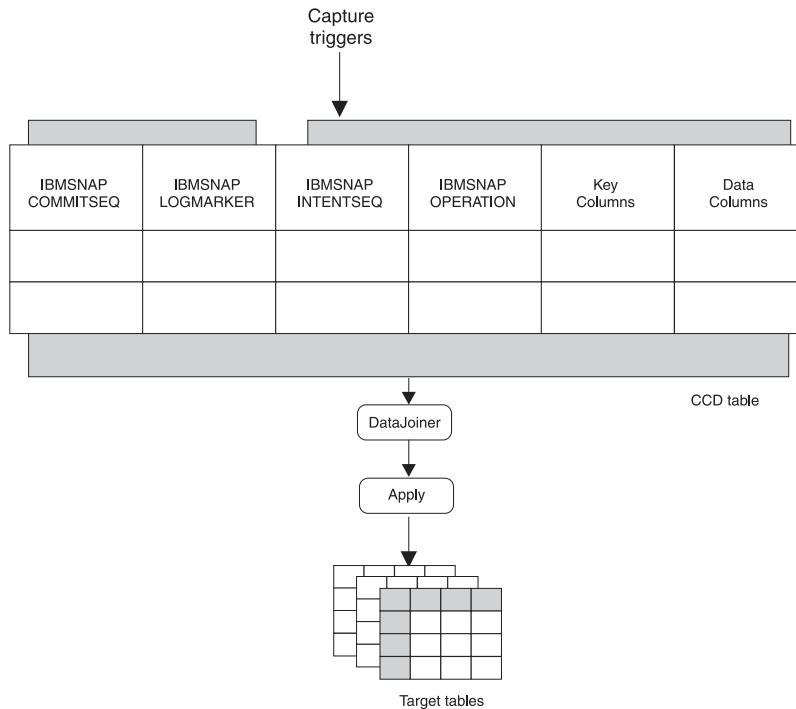


Figure 60. The CCD Table. The Apply program uses information in the CCD table to determine what to replicate.

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* 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* 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* 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* 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 61 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 61. 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 266 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                                                                                                                      |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Insert</b> | The before-image column has a NULL value.                                                                                         |
| <b>Update</b> | Column values before the change are captured in the before-image columns; values after the change are in the after-image columns. |
| <b>Delete</b> | 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 266 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. See “Replication Logical Partitioning Key Support” in Chapter 2 of *IBM DB2 Replication Guide and Reference: V5* to learn how to use this option. 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 62 on page 262 and Figure 63 on page 263 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.

**Push Scenario**

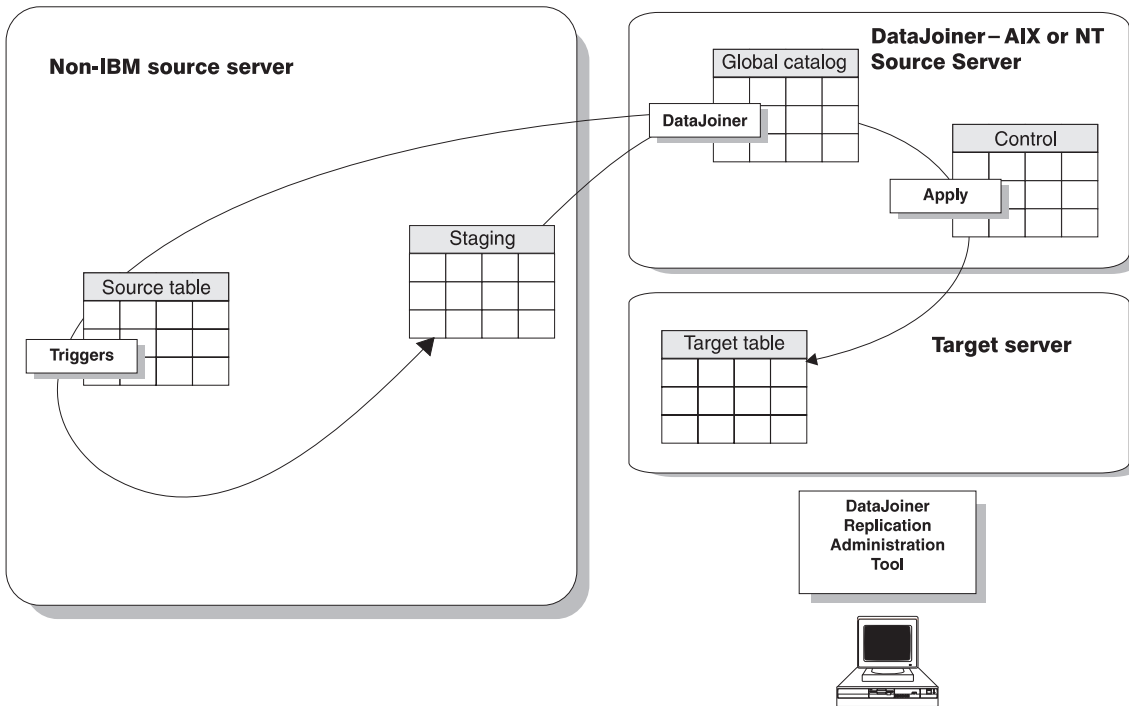


Figure 62. The Apply Program Push Scenario

## Pull Scenario

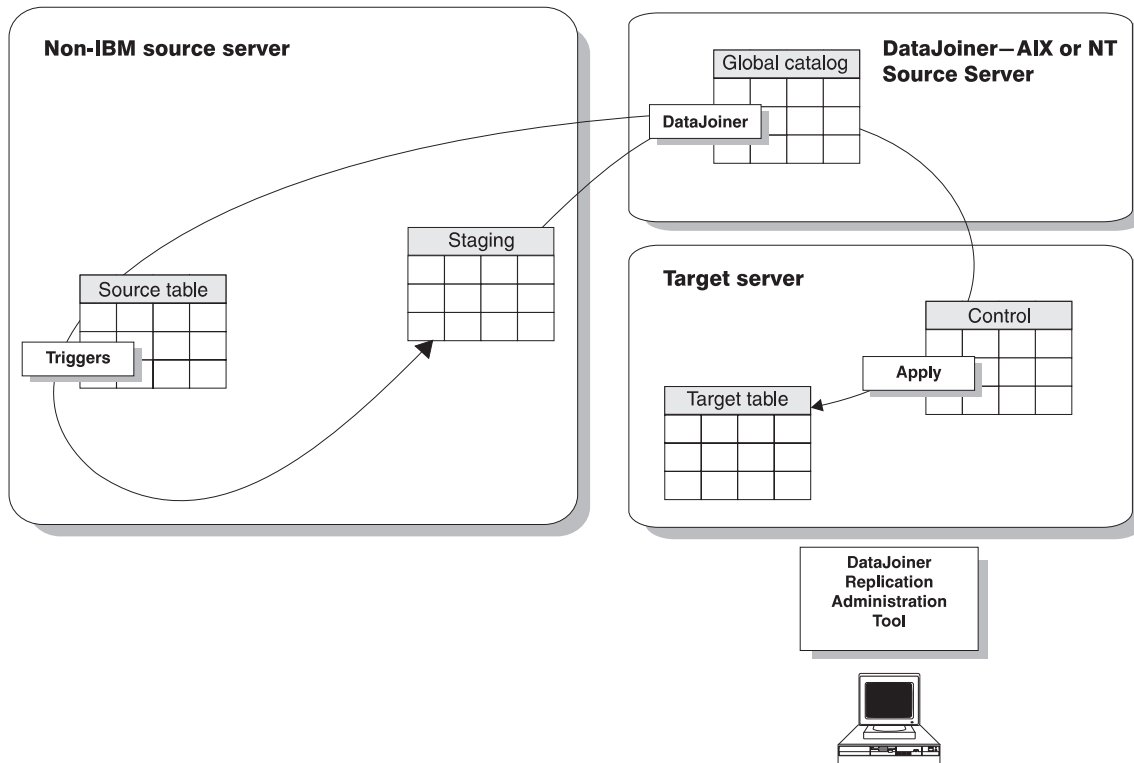


Figure 63. The Apply Program Pull Scenario

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* 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* 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 277 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 7 on page 266 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 7. 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 <b>Promote Subscriptions</b> 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"><li>Apply Qualifier</li><li>Set Name</li><li>Source server</li><li>Source alias</li><li>Target server</li><li>Target alias</li><li>Control server</li><li>Control alias</li></ul> |

## 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 27. 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 8:

*Table 8. Supported Sources and Targets*

| Database                                                                                      | Capture and Apply Programs needed                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DB2 UDB V5                                                                                    | Includes V5 Capture and Apply programs. The V5 Capture program is required when DB2 UDB is used as source. The V5 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                                   | The V5 Capture program for MVS is required when DB2 for MVS or DB2 for OS/390 is the source. The V5 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 V5 Capture program is required when DB2 for common servers or DataJoiner is the source. The V5 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 (AIX) and Oracle V7.3 or later (Windows NT)                           | The replication administration tool creates Capture triggers on an Oracle source. The V5 Apply program is required when Oracle is the target (with DataJoiner as the target server).                                 |
| Informix V7.1 or later (AIX) and Informix V7.2 or later (Windows NT)                          | The replication administration tool creates Capture triggers on an Informix source. The V5 Apply program is required when Informix is the target (with DataJoiner as the target server).                             |
| Sybase V4.6 or later (AIX) and Sybase V11 or later (Windows NT)                               | The replication administration tool creates Capture triggers on an Sybase source. The V5 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 V5 Apply program is required when SQL Anywhere is the target (with DataJoiner as the target server).                     |
| Microsoft SQL Server V4.21 or later (AIX) and Microsoft SQL Server V6.0 or later (Windows NT) | The replication administration tool creates Capture triggers on an Microsoft SQL Server source. The V5 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 27 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
- 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 64 on page 271 shows software requirements if DataJoiner for Windows NT and the replication administration tool are located on the same workstation. Figure 65 on page 271 and Figure 66 on page 272 show software requirements if DataJoiner and the replication administration tool are on separate systems.

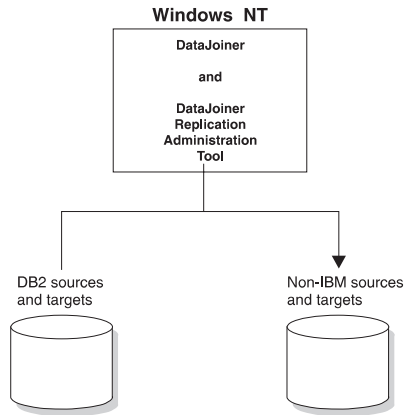


Figure 64. DataJoiner and the Replication Administration Tool on a Single Windows NT Workstation.

In Figure 64, 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

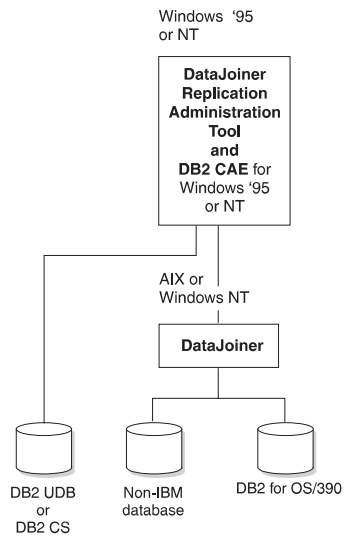
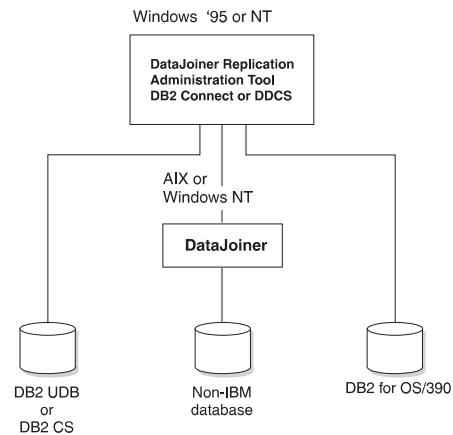


Figure 65. DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses DataJoiner to Connect.

In Figure 65 on page 271, the replication administration tool installed on Windows NT or '95 works with DataJoiner for AIX 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 or Windows NT V3.5.1 or later with Microsoft service pack 3
- Client Application Enablers (CAE) V2.11 or later for Windows NT or Windows 95

DataJoiner V2 for AIX 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 66. DataJoiner and the Replication Administration Tool on Separate Systems: DB2 for OS/390 Uses the Replication Administration Tool to Connect.*

In Figure 66, the replication administration tool provides access to DB2 for OS/390. With this configuration, the replication administration tool software requirements are:

- Microsoft Windows 95 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 AIX 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 DPROG for Microsoft Jet Software Requirements

To run DB2 DPROG for Microsoft Jet, the following software must be installed on the client machine:

- Microsoft Windows NT or Windows 95
- Microsoft Jet or Microsoft Access for Windows NT (or for Windows 95)
- DB2 Client Application Enabler (DB2 CAE)
- DB2 DPROG 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 DPROG 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 AIX is provided with DataJoiner for AIX. 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*. Table 9 lists the Capture and Apply programs supported by DataJoiner with the replication administration tool.

*Table 9. Summary of Capture and Apply Programs Supported by DataJoiner with the Replication Administration Tool*

| The Capture Program                 | The Apply Program                                                                  |
|-------------------------------------|------------------------------------------------------------------------------------|
| Capture for AIX                     | Apply for AIX (you can select Apply for AIX 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                                                                     |
| Capture for UNIX OS-based platforms | Apply for UNIX OS-based platforms                                                  |

---

## 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 276.

### 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.<sup>9</sup>
- 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 289 describes how to set up one or more

---

9. In Oracle only.



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 277.

## 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* for Capture program authorization and security issues.

To complete the set up of your replication environment you need to map your authorizations through DataJoiner. See “Mapping of User IDs” on page 277.

## Authorization Requirements for the Apply Program

The Apply program user ID must be a valid logon ID on the source, control, and target servers. The user ID that runs the Apply program 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 AIX” on page 293 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 277.

## 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 AIX**

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 265. 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 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.<sup>10</sup>

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 67 on page 278 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.

---

10. If you are using Microsoft SQL Server in your replication environment, do not use alias user IDs. Alias user IDs will not map correctly.

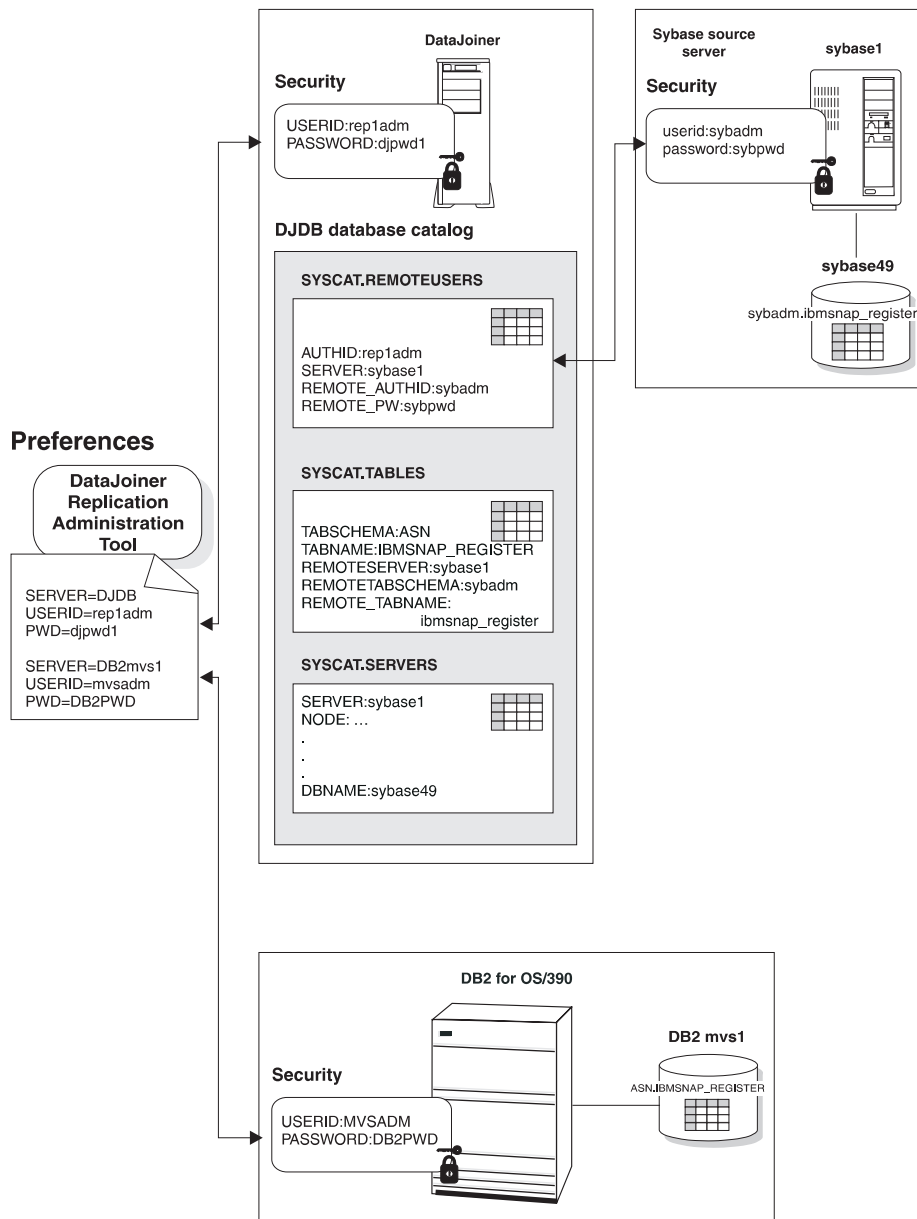


Figure 67. DataJoiner Maps the Replication Administration Tool User IDs to Source and Target. The user ID specified in the replication administration tool is mapped in DataJoiner to a non-IBM source and a DB2 for OS/390 target

In Figure 67, 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 68 on page 280 shows what happens when the Apply program performs replication.

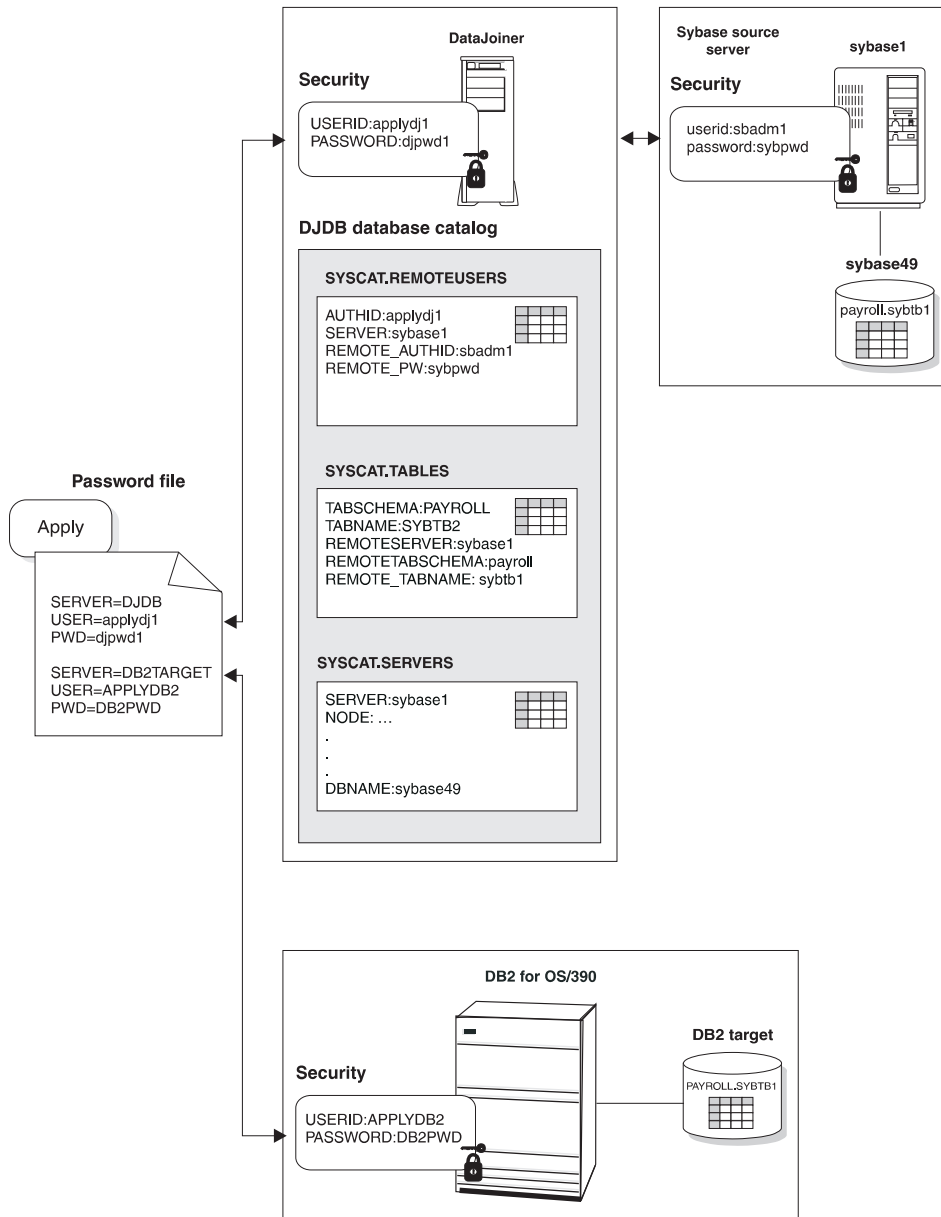


Figure 68. DataJoiner Uses the Apply Password File to Connect to Source and Target. The Apply program password file identifies the user ID and password DataJoiner uses to access a non-IBM source and a DB2 for OS/390 target.

In Figure 68, the subscription set specifies the source, target and Apply qualifier. 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 28. 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 AIX. After this chapter, you must proceed to “Chapter 29. Installing the Replication Administration Tool and Connecting All Databases” on page 287 to install the replication administration tool and connect all databases that are involved in replication.

---

### Replication in DataJoiner for AIX

This section describes how to prepare DataJoiner for AIX for replication.

#### Installing DataJoiner

Restore product files using `installp` or `smitty` by following the steps in “Chapter 4. Installing DataJoiner” on page 35. Installation of some components is optional. Make sure that you install:

- TCP/IP support, if the replication administration tool will access DataJoiner through TCP/IP
- APPC support, if the replication administration tool will access DataJoiner through advanced program-to-program communications (APPC)
- DRDA support, if accessing DB2 for MVS as a source or target server
- The Apply component of DataJoiner

Later, as described in “Chapter 29. Installing the Replication Administration Tool and Connecting All Databases” on page 287, you will install the replication administration tool on your NT workstation.

## Setting up an Instance

In “Chapter 5. Setting up an Instance” on page 55, you:

- Create an instance
- Set up environment variables
- Create DataJoiner databases
- Set up the Apply for AIX as a local client

This section focuses on setting up the Apply program and creating DataJoiner databases as pertaining to replication.

### Setting up Apply for AIX as a Local Client

In “Step 2: Set Environment Variables” on page 57, identify the Apply user ID as a local client on DataJoiner.

In addition to using the Apply for AIX that is provided with DataJoiner, you can also use other Apply programs on other platforms as discussed in “Connecting Clients to DataJoiner” on page 285.

### Creating DataJoiner Databases

In “Step 5: Create a DataJoiner Database” on page 61, you create databases that will be configured to access non-IBM databases as part of your replication system. 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.

#### For Oracle as a source:

Use the COLLATE USING parameter within the CREATE DATABASE command. Use IDENTITY.

## Configuring Database Connections

Within “Chapter 6. Configuring Access to Data Sources” on page 69, several sections are applicable to replication. The Apply program in DataJoiner uses the connections that you configure in this section to access source, target, and control servers.

*Table 10. Configuring Database Connections in DataJoiner: AIX*

| If Accessing            | You need to               | Refer to this section                                           |
|-------------------------|---------------------------|-----------------------------------------------------------------|
| DB2 for MVS             | Catalog DB2 in DataJoiner | “Accessing DRDA Data Sources using DataJoiner DDCS” on page 95. |
| DB2 UDB V5 or DB2 CS V2 | Catalog DB2 in DataJoiner | “Chapter 8. Accessing DB2RA Data Sources” on page 101.          |

Table 10. Configuring Database Connections in DataJoiner: AIX (continued)

| If Accessing            | You need to                                                                                                | Refer to this section                                                                                                                                                                                                               |
|-------------------------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Informix                | Create a server mapping, user mapping, and nickname for Informix, within DataJoiner.                       | "Chapter 9. Accessing Informix Data Sources" on page 131.                                                                                                                                                                           |
| MS SQL Server or Sybase | Create a server mapping, user mapping, and nickname for Microsoft SQL Server or Sybase, within DataJoiner. | "Chapter 10. Accessing Microsoft SQL Server and Sybase Data Sources Using Sybase Open Client" on page 143, or, if applicable, refer to "Chapter 11. Accessing Microsoft SQL Server through the Visigenic ODBC Driver" on page 149 . |
| Oracle                  | Create a server mapping, user mapping, and nickname for Oracle, within DataJoiner.                         | "Chapter 12. Accessing Oracle Data Sources Using SQL*Net or Net8" on page 155.                                                                                                                                                      |

You must define server and user mappings 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 *target* servers with one DataJoiner database. The DataJoiner databases that you set up reside on one DataJoiner instance.

## Connecting Clients to DataJoiner

"Part 4. Connecting Clients to DataJoiner" on page 173, applies to Apply programs installed on DB2 for MVS, DB2 UDB, or DB2 CS that must access non-IBM sources or targets through DataJoiner.

For Apply for MVS, refer to "Chapter 21. Configuring the DataJoiner DRDA Application Server and Clients" on page 207. The chapter describes how to enable applications on MVS, such as Apply for MVS, to access DataJoiner.

For Apply programs on DB2 UDB or DB2 CS, refer to the chapter applicable to your network's communication protocol: TCP/IP, IPX/SPX, or APPC.

The steps that are described in "Part 4. Connecting Clients to DataJoiner" on page 173 must be performed for each DataJoiner database that is created to replicate with a remote Apply program. Later, in "Configuring Access from the Replication Administration Tool to DataJoiner and DB2" on page 288, you will connect the replication administration tool as a client to DataJoiner.

## What to Do Next

Before proceeding to "Chapter 29. Installing the Replication Administration Tool and Connecting All Databases" on page 287, 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 273 for a list that is supported by DataJoiner with the replication administration tool. And refer to the *IBM DB2 Replication Guide and Reference: V5* for instructions on installing the Capture and Apply programs.

---

## Chapter 29. 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 or NT workstation.
- Configure access from the workstation on which the replication administration tool is installed to each source, target, and control server.
- Set the DB2CODEPAGE environment variable for DataJoiner for AIX access.
- 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 OBJRexx as part of the installation. Check to see if you already have OBJRexx installed and verify the release level. The replication administration tool will replace your downlevel copy of OBJRexx, or you can choose not to install OBJRexx 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, no administrator log on is required. To install the replication administration tool:

1. Insert the DataJoiner CD-ROM into the appropriate drive.
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 69.

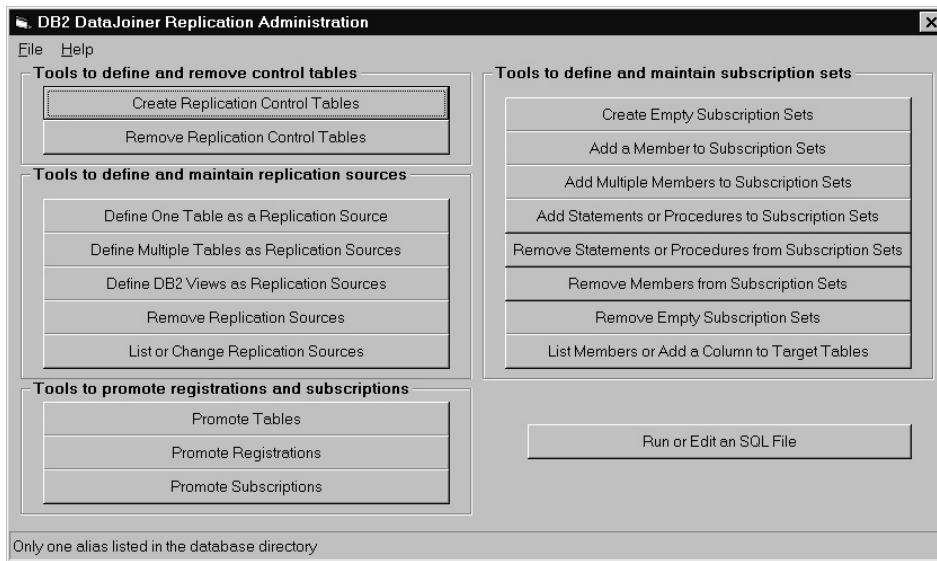


Figure 69. The Replication Administration Tool Primary Window

## 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 "Part 4. Connecting Clients to DataJoiner" on page 173. Refer to the chapter applicable to your network protocol: TCP/IP, IPX/SPX, APPC, or NetBIOS.

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 21. Configuring the DataJoiner DRDA Application Server and Clients” on page 207, or your *DDCS User’s Guide*, V2.3.

---

## Setting DB2CODEPAGE for DataJoiner for AIX Access

From your replication administration tool workstation, set the DB2CODEPAGE environment variable when the replication administration tool access DataJoiner for AIX. The value you set is based on your country code. For example, if your country code is US, you would:

1. Select the **My Computer** icon.
2. Select **Select System** icon
3. From the System Properties folder, select **Environment**.
4. Type in DB2CODEPAGE in **Variable**.
5. For US, type in the value 437 in **User Variables**.

---

## Setting Administrative Preferences

You can specify your preferences for:

- Location of console or file output
- User IDs and passwords
- Tracing SQL execution activity

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.

| 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 70 on page 290 shows the Connection page on the Preferences notebook.



Figure 70. 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 and Windows 95 workstations you must bind DB2 utilities to each DataJoiner and each DB2 database from at least one Windows 95 workstation and at least one Windows NT workstation. An example of what you enter is:

```
cd c:\sqllib\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 71 on page 291.



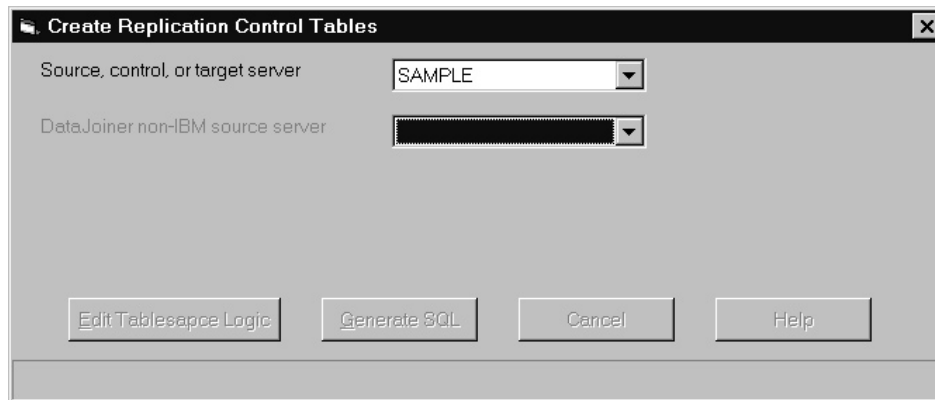


Figure 71. Creating Replication Control Tables.

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 pull-down 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 are:

- TS\_UOW for the UOW table
- TS\_CNTL for all other control tables

**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 297

page 295. 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 290.

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 AIX”. For more information on binding other Capture and Apply programs, refer to the *IBM DB2 Replication Guide and Reference: V5*.

---

## Finalizing the Setup for the Apply Program in AIX

This section describes how to do the following tasks in the AIX environment:

- Bind the Apply program.
- Provide end-user authentication.

### Binding the Apply Program in AIX

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 AIX system. 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 274.
2. Logon to the AIX 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/sql/lib/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
db2 bind @applyur.lst isolation ur blocking all
```

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 the packages 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*.

## Providing End-User Authentication in AIX

Remote DB2 systems involved in replication might require that the Apply program running locally on DataJoiner for AIX 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.

For more information about authentication and security, refer to the *IBM DataJoiner Administration Supplement* or the *IBM DB2 Universal Database Administration Guide*.

### If you create a password file:

The password file must:

- Be named:

```
APPLYQUALinstnameCNTLSRVR.PWD
```

Where:

*APPLYQUAL*

The Apply qualifier, in uppercase.

*instname*

The DataJoiner instance name, in lowercase, that Apply program runs under.  
The value of DB2INSTANCE.

*CNTLSRVR*

The name of the control server, in uppercase

For example: DATADIRApply1REPRDDB.PWD.

The naming convention that is used here is the same as the log file name (.LOG) and the spill file name (.SPL), 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 use 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.

We recommend that you limit read access of this file to the user ID that runs the Apply program.

**If you do not create a password file:**

The Apply program in DataJoiner must be able to issue an SQL CONNECT statement without specifying the user ID and password. Ensure that:

- The DB2 for MVS database is cataloged as AUTHENTICATION=CLIENT.
- The logon ID belongs to PRIMARY GROUP=SYSTEM.

When copying from DB2 for MVS sources, ensure that:

- SECURITY=SAME for MVS CPI-C node.
- You specify the following values when you define the LU name that uses the VTAM APPL:
  - VERIFY=NONE to indicate that any logical unit (LU) can request an LU-LU session.
  - SECACPT=ALREADYV to indicate user ID and password checking at the requester.

---

## Chapter 30. 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 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 72.

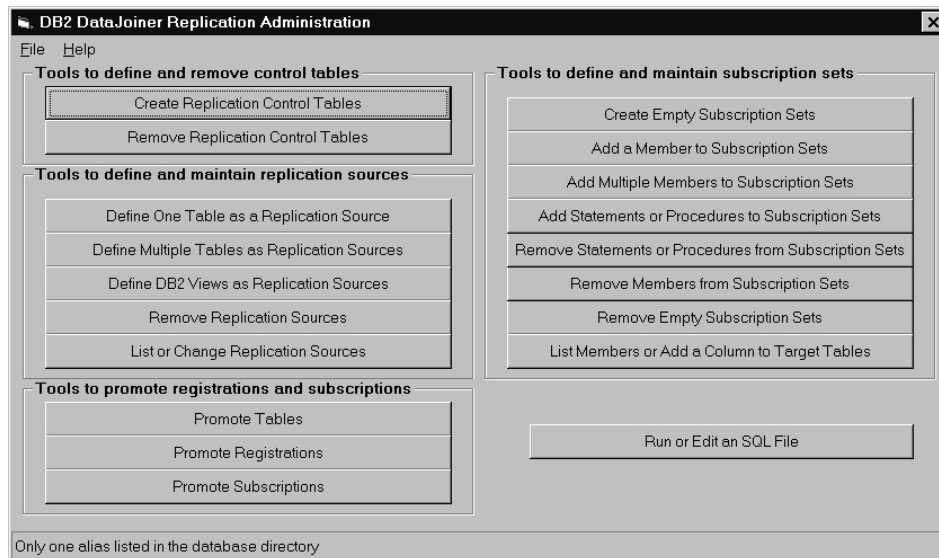


Figure 72. The Replication Administration Tool Primary Window

---

### 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” 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.
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 edit the SQL. See “Editing the Replication Administration Tool-Generated SQL” on page 297 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 298 for more information.

## Editing the Replication Administration Tool Logic

Three files are available for edit: SRCESVR.REX, TARGSVR.REX, and CNTRLSVR.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.

### CNTRLSVR.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 CNTRLSVR.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 Create 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 add one or multiple members to subscription sets and click **Edit Create 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.

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 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 Capture and Apply programs. For more information on running Capture and Apply programs, refer to the *IBM DB2 Replication Guide and Reference: V5*. Capture triggers begin running automatically.



---

## Chapter 31. 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 11. Data Type Conversion: DB2 to Oracle*

| DB2 for MVS<br>Source data type | Oracle Target<br>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 12. Data Type Conversion: DB2 to Informix*

| DB2 for MVS<br>Source data type | Informix Target<br>data type | Changes after Create Nickname |        |                                              |
|---------------------------------|------------------------------|-------------------------------|--------|----------------------------------------------|
|                                 |                              | COLTYPE                       | LENGTH | SCALE                                        |
| FOR BIT DATA(n)                 | CHAR(n) <sup>1</sup>         |                               |        |                                              |
| CHAR(n)                         | CHAR(n)                      |                               |        |                                              |
| VARCHAR(<256)                   | VARCHAR(n)                   |                               |        |                                              |
| VARCHAR(>=256)                  | TEXT                         |                               |        |                                              |
| GRAPHIC(n)                      | BYTE                         |                               |        |                                              |
| VARGRAPHIC(n)                   | BYTE                         |                               |        | No updates occur<br>after create<br>nickname |
| LONG VARCHAR                    | TEXT                         |                               |        |                                              |
| DATE                            | DATE                         |                               |        |                                              |
| TIMESTAMP                       | DATETIME                     |                               |        |                                              |
|                                 | YEAR TO                      |                               |        |                                              |

Table 12. Data Type Conversion: DB2 to Informix (continued)

| DB2 for MVS<br>Source data type | Informix Target<br>data type  | COLTYPE | Changes after Create Nickname<br>LENGTH | SCALE |
|---------------------------------|-------------------------------|---------|-----------------------------------------|-------|
|                                 | FRACTION(5)                   |         |                                         |       |
| TIME                            | DATETIME<br>HOUR TO<br>SECOND |         |                                         |       |
| SMALLINT                        | SMALLINT                      |         |                                         |       |
| INTEGER                         | INT                           |         |                                         |       |
| DECIMAL(n,m)                    | DECIMAL(n,m)                  |         |                                         |       |
| FLOAT                           | FLOAT                         |         |                                         |       |

**Note:** <sup>1</sup>For 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 13. Data Type Conversion: DB2 to MS SQL Server, Sybase, or SQL Anywhere

| DB2 for MVS<br>SOURCE         | MS SQL<br>SERVER,<br>SYBASE, OR<br>SQL ANYWHERE<br>TARGET | COLTYPE           | Changes after Create Nickname<br>LENGTH | SCALE |
|-------------------------------|-----------------------------------------------------------|-------------------|-----------------------------------------|-------|
| CHAR (n) FOR<br>BIT DATA      | BINARY(n)                                                 | CHAR <sup>1</sup> |                                         |       |
| VARCHAR(n)<br>FOR BIT DATA(N) | VARBINARY(n) or<br>BINARY(n)                              |                   |                                         |       |
| CHAR(n)                       | CHAR(n)                                                   |                   |                                         |       |
| VARCHAR(<256)                 | VARCHAR2(n) or<br>VARCHAR(n)                              |                   |                                         |       |
| VARCHAR(<256)                 | TEXT                                                      |                   |                                         |       |
| GRAPHIC(n) <sup>2</sup>       | ?                                                         |                   |                                         |       |
| VARGRAPHIC(n)<br><sup>2</sup> | ?                                                         |                   |                                         |       |
| LONG VARCHAR<br><sup>2</sup>  | ?                                                         |                   |                                         |       |
| DATE                          | DATETIME or<br>DATE                                       | DATE              | 4                                       |       |
| TIMESTAMP                     | DATETIME or<br>DATESTAMP                                  |                   |                                         |       |
| TIME                          | DATETIME or<br>TIME                                       | TIME              | 3                                       |       |
| SMALLINT                      | SMALLINT                                                  |                   |                                         |       |
| INTEGER                       | INT                                                       |                   |                                         |       |

Table 13. 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 |
| DECIMAL(n,m)       | DECIMAL(n,m)                                  |                               |        |       |
| FLOAT              | FLOAT                                         |                               |        |       |

**Note:** <sup>1</sup>The 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:** <sup>2</sup>These 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 14. 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 <sup>1</sup>                                                |                               |        |       |
| VARCHAR(n) FOR BIT DATA(N) | Memo or OLE Object <sup>1</sup>                                                |                               |        |       |
| CHAR(n)                    | Text or Memo <sup>1</sup>                                                      |                               |        |       |
| VARCHAR(<256)              | Text or Memo <sup>1</sup>                                                      |                               |        |       |
| GRAPHIC(n) <sup>2</sup>    | ?                                                                              |                               |        |       |
| VARGRAPHIC(n) <sup>2</sup> | ?                                                                              |                               |        |       |
| LONG VARCHAR <sup>2</sup>  | Memo                                                                           |                               |        |       |
| 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 <sup>1</sup> |                               |        |       |
| PRECISION, FLOAT, DOUBLE   | Number (Double)                                                                |                               |        |       |

**Note:** <sup>1</sup>The value used depends on the precision set for the field.

**Note:** <sup>2</sup>These DB2/MVS data types have not been tested.





---

## Chapter 32. Non-IBM Databases: Table Names, Control Tables, and the CCD

This section describes how the CCD table name in a non-IBM database and the associated nickname are determined when a replication source is defined.

---

### 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 current SQLID 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 Control Table

The `ASN.IBMSNAP_REGISTER` nickname refers to a table in the non-IBM database with the same unqualified name with probably a different qualifier, for example, `MYUSERID.IBMSNAP_REGISTER`. Even though the register control table resides 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.



---

## 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)

Because DRDA Application Servers and Application Requesters communicate using the APPC communication protocol, the SNA support option (`djx_02_01.sna_clients`) must also be installed.

---

### Supported DRDA Functions

In DRDA, a function is categorized as either required or optional. Table 15 identifies which functions (required and optional) are implemented in the AS on DataJoiner, and Table 16 on page 308 lists the bind option values that are supported.

*Table 15. DataJoiner DRDA Application Server Function*

| Description                    | Required (R)<br>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 16 on page 308 shows the BIND options that are supported by the DataJoiner DRDA Application Server.

Table 16. 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>                      | Yes        | VERSION                                |                              |                                                                  | VERSION                      |
|                                      | Any other value                  | No         |                                        |                              |                                                                  |                              |
| Bind Existence Checking              | <u>Object existence optional</u> | No         | VALIDATE ( <u>RUN</u> ) <sup>b</sup>   | <u>NOEXIST</u>               | GENLVL( <u>10</u> , 11-40)                                       | VALIDATE <u>RUN</u>          |
|                                      | Object existence required        | Yes        | VALIDATE (BIND) <sup>p</sup>           | EXIST                        | GENLVL(00-09)                                                    | VALIDATE BIND                |
| Package Replacement Option           | <u>Replacement allowed</u>       | Yes        | 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([...] * <u>SYSVAL</u> ) (Note 6) | DECDEL PERIOD                |
|                                      | Comma                            | No         | COMMA                                  | COMMA                        | OPTION([...] *COMMA) or OPTION([...] * <u>SYSVAL</u> ) (Note 6)  | DECDEL COMMA                 |
| 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                 |

Table 16. 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     |
|-----------------------------------|--------------------------------------|--------------|--------------------------------------------|------------------------------|-----------------------------------------------------|------------------------------|
| Time Format (Note 7)              | <b>ISO</b>                           | Yes          | TIME(ISO)<br>(Note 8)                      | TIME( <b>ISO</b> )           | TIMFMT(*ISO)<br>(Note 8)                            | DATETIME <b>ISO</b> (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( <b>RR</b> ) <sup>b</sup>        | ISOLATION( <b>RR</b> )       |                                                     | ISOLATION RR                 |
|                                   | Read Stability (All)                 | Yes          |                                            | ISOLATION(RS)                | COMMIT(*ALL)                                        | ISOLATIONRS                  |
|                                   | Cursor stability                     | Yes          | ISOLATION(CS) <sup>b</sup>                 | ISOLATION(CS)                | COMMIT(*CS)                                         | ISOLATION <b>CS</b>          |
|                                   | Uncommitted Read (Change)            | Yes          |                                            | ISOLATION(UR)                | COMMIT(* <b>CHG</b> )                               | ISOLATION UR                 |
|                                   | No commit                            | No (Note 11) |                                            |                              | COMMIT(*NONE)                                       | ISOLATION NC                 |
| Bind Creation Control             | <b>No errors allowed</b>             | Yes          | SQLERROR ( <b>NOPACKAGE</b> ) <sup>b</sup> | <b>NOCHECK</b>               | OPTION([...] *GEN) GENLVL(00-09, <b>10</b> , 11-20) | SQLERROR <b>NOPACKAGE</b>    |
|                                   | Check only                           | Yes          |                                            | CHECK                        | OPTION([...] *NOGEN)                                | SQLERROR CHECK               |
|                                   | Errors allowed                       | No           | SQLERROR (CONTINUE) <sup>b</sup>           | ERROR                        | OPTION([...] *GEN) GENLVL(21-40)                    | SQLERROR CONTINUE            |
| Bind Explain Option               | <b>No SQL statements</b>             | Yes          | EXPLAIN( <b>NO</b> ) <sup>b</sup>          | EXPLAIN( <b>NO</b> )         |                                                     | EXPLAIN <b>NO</b>            |
|                                   | All explainable SQL statements       | No           | EXPLAIN(YES) <sup>b</sup>                  | EXPLAIN(YES)                 |                                                     | EXPLAIN YES                  |
| Package Owner Identifier          | < <b>Authorization ID</b> >          | Yes          | OWNER <sup>b</sup>                         | OWNER                        |                                                     | OWNER                        |
|                                   | Any other value                      | No           |                                            |                              |                                                     |                              |
| RDB Release Option                | <b>Release at commit</b>             | Yes          | RELEASE ( <b>COMMIT</b> ) <sup>b</sup>     | RELEASE ( <b>COMMIT</b> )    |                                                     | RELEASE <b>COMMIT</b>        |
|                                   | Release at conversation deallocation | No           | RELEASE (DEALLOCATE) <sup>b</sup>          | RELEASE (DEALLOCATE)         |                                                     | RELEASE DEALLOCATE           |
| Default RDB Collection ID         | < <b>Authorization ID</b> >          | Yes          | QUALIFIER <sup>b</sup>                     | QUALIFIER                    | DFTRDBCOL                                           | QUALIFIER                    |

Table 16. 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  |
|-------------------------------|------------------------------------------------------|------------|----------------------------------------|-----------------------------------------------|--------------------------|---------------------------|
|                               | Any other value                                      | No         |                                        |                                               |                          |                           |
| Title (Package Description)   | Any value (ignored by DB2)                           | Yes        |                                        | LABEL                                         | TEXT                     | TEXT                      |
| Query Block Protocol Control  | <b>Fixed row</b>                                     | Yes        | CURRENTDATA (YES) <sup>b</sup>         | SBLOCK                                        | ALWBLK(*READ)            | BLOCKING UNAMBIG          |
|                               | Limited block                                        | Yes        | CURRENTDATA (NO) <sup>p</sup>          | BLOCK                                         | ALWBLK (*ALLREAD)        | BLOCKING ALL              |
|                               | Forced fixed row                                     | Yes        |                                        | <b>NOBLOCK</b>                                | ALWBLK(*NONE)            | BLOCKING NO               |
| Package Default Char. Subtype |                                                      |            |                                        |                                               |                          |                           |
|                               | <b>Use system default</b>                            | Yes        |                                        |                                               |                          | CHARSUB DEFAULT           |
| If Default CCSID is SBCS      | BIT                                                  | No         |                                        | CHARSUB(BIT)                                  |                          | CHARSUB BIT               |
| If Default CCSID is SBCS      | SBCS                                                 | Yes        |                                        | 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                                                 | Yes        |                                        | CHARSUB(MBCS)                                 |                          | CHARSUB MBCS              |
|                               | Any other value                                      | No         |                                        |                                               |                          |                           |
| Package Default CCSID         | <b>Value specified when DB2 database was created</b> | Yes        |                                        | CCSIDSBCS()<br>CCSIDGRAPHIC()<br>CCSIDMIXED() |                          | CCSID<br>CCSIDG<br>CCSIDM |
|                               | Any other value                                      | No         |                                        |                                               |                          |                           |
| Decimal Precision (Note 12)   | 31                                                   | Yes        | DEC(31)                                |                                               |                          | DEC 31                    |
|                               | Any other value                                      | No         | DEC(15)                                |                                               |                          | DEC 15                    |
| Replaced Package Version Name | <b>Null</b>                                          | Yes        | REPLVER <sup>b</sup>                   |                                               |                          | REPLVER                   |
|                               | Any other value                                      | No         |                                        |                                               |                          |                           |

Table 16. 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 <sup>b</sup>.
- (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 15 on page 307 and Table 16 on page 308 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 17, which follows.

- **AIX 3.2 Locale** and **AIX 4.1 Locale** list the locale values that are supported by the database manager. The locale specifies a language\_territory combination. The value of territory is used to map to the country code.
- **Country Name** specifies the name of the country. Where required, the language for that locale is listed in parentheses ().
- **Country Code** shows the country code that is mapped from the territory portion of the locale, and which is used by the database manager internally for providing country-specific support.
- **Supported Code Set** shows the code set that is associated with the supported locale. The code set is mapped to the DB2 code page.
- **Database Code Page** shows the IBM-defined code page as mapped from the operating system code set.
- **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 17. CCSID Support for DataJoiner for AIX DRDA Application Server

| AIX 3.2 Locale | AIX 4.1 Locale | Country Name     | Country Code | Supported Code Set | Database Code Page | EBCDIC AR CCSID                                  | ASCII AR CCSID                   |
|----------------|----------------|------------------|--------------|--------------------|--------------------|--------------------------------------------------|----------------------------------|
| ar_AA          | ar_AA          | Arabic countries | 785          | ISO8859-6          | 1089 <sup>a</sup>  | n/a                                              | 864 1046 1089 1256               |
| Ar_AA          | Ar_AA          | Arabic countries | 785          | IBM-1046           | 1046               | 420                                              | 864 1046 1089 1256               |
| C              | C              | Australia        | 001          | IBM-850            | 850                | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| n1_BE          | n1_BE          | Belgium (Dutch)  | 032          | ISO-8859-1         | 819                | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| N1_BE          | N1_BE          | Belgium (Dutch)  | 032          | IBM-850            | 850                | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |

Table 17. CCSID Support for DataJoiner for AIX DRDA Application Server (continued)

| AIX 3.2<br>Locale | AIX 4.1<br>Locale | Country<br>Name     | Country<br>Code | Supported<br>Code Set | Database<br>Code Page | EBCDIC AR<br>CCSID                               | ASCII AR<br>CCSID                    |
|-------------------|-------------------|---------------------|-----------------|-----------------------|-----------------------|--------------------------------------------------|--------------------------------------|
| fr_BE             | fr_BE             | Belgium<br>(French) | 032             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| Fr_BE             | Fr_BE             | Belgium<br>(French) | 032             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| bg_BG             | bg_BG             | Bulgaria            | 359             | ISO-8859-5            | 915                   | 1025                                             | 855 866 915<br>1251 1283             |
| en_US             | en_US             | Canada<br>U.S.A.    | 001             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| En_US             | En_US             | Canada<br>U.S.A.    | 001             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| fr_CA             | fr_CA             | Canada<br>(French)  | 002             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>863 1051 1252<br>1275 |
| Fr_CA             | Fr_CA             | Canada<br>(French)  | 002             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>863 1051 1252<br>1275 |
| hr_HR             | hr_HR             | Croatia             | 385             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282                 |
| cs_CS             | cs_CZ             | Czech<br>Republic   | 042             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282                 |
| da_DK             | da_DK             | Denmark             | 045             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| Da_DK             | Da_DK             | Denmark             | 045             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |
| fi_FI             | fi_FI             | Finland             | 358             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275     |

Table 17. CCSID Support for DataJoiner for AIX DRDA Application Server (continued)

| AIX 3.2<br>Locale | AIX 4.1<br>Locale | Country<br>Name  | Country<br>Code | Supported<br>Code Set | Database<br>Code Page | EBCDIC AR<br>CCSID                               | ASCII AR<br>CCSID                |
|-------------------|-------------------|------------------|-----------------|-----------------------|-----------------------|--------------------------------------------------|----------------------------------|
| Fi_FI             | Fi_FI             | Finland          | 358             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| mk_MK             | mk_MK             | FYR<br>Macedonia | 389             | ISO8859-5             | 915                   | 1025                                             | 855 866 915<br>1251 1283         |
| fr_FR             | fr_FR             | France           | 033             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Fr_FR             | Fr_FR             | France           | 033             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| de_DE             | de_DE             | Germany          | 049             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| De_DE             | De_DE             | Germany          | 049             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| e1_GR             | e1_GR             | Greece           | 030             | ISO8859-7             | 813                   | 423 875                                          | 813 869 1253<br>1280             |
| hu_HU             | hu_HU             | Hungary          | 036             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| is_IS             | is_IS             | Iceland          | 354             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Is_IS             | Is_IS             | Iceland          | 354             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| iw                | iw_IL             | Israel           | 972             | ISO8859-8             | 916                   | 424                                              | 862 916 1255                     |
| it_IT             | it_IT             | Italy            | 039             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| It_IT             | It_IT             | Italy            | 039             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |



Table 17. CCSID Support for DataJoiner for AIX DRDA Application Server (continued)

| AIX 3.2<br>Locale | AIX 4.1<br>Locale | Country<br>Name        | Country<br>Code | Supported<br>Code Set | Database<br>Code Page | EBCDIC AR<br>CCSID                               | ASCII AR<br>CCSID                |
|-------------------|-------------------|------------------------|-----------------|-----------------------|-----------------------|--------------------------------------------------|----------------------------------|
| ja_JP             | ja_JP             | Japan                  | 081             | IBM-eucJP             | 954                   | 930 931 939<br>5026 5035                         | 932 942 943<br>954               |
| Ja_JP             | Ja_JP             | Japan                  | 081             | IBM-932               | 932                   | 930 931 939<br>5026 5035                         | 932 942 943<br>954               |
| ko_KR             | ko_KR             | Korea                  | 082             | IBM-eucKR             | 970                   | 933                                              | 949 970                          |
| C                 | C                 | Latin<br>America       | 001             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| n1_NL             | n1_NL             | Netherlands<br>(Dutch) | 031             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| N1_NL             | N1_NL             | Netherlands<br>(Dutch) | 031             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| no_NO             | no_NO             | Norway                 | 047             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| No_NO             | No_NO             | Norway                 | 047             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| zh_CN             | zh_CN             | PRC (China)            | 086             | IBM-eucCN             | 1383                  | 935                                              | 1381 1383                        |
| p1_PL             | p1_PL             | Poland                 | 048             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| pt_PT             | pt_PT             | Portugal               | 351             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Pt_PT             | Pt_PT             | Portugal               | 351             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| ro_RO             | ro_RO             | Romania                | 040             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| ru_RU             | ru_RU             | Russia                 | 007             | ISO8859-5             | 915                   | 1025                                             | 855 866 915<br>1251 1283         |
| sp_YU             | sr_SP             | Serbia<br>(Cyrillic)   | 381             | ISO8859-5             | 915                   | 1025                                             | 855 866 915<br>1251 1283         |

Table 17. CCSID Support for DataJoiner for AIX DRDA Application Server (continued)

| AIX 3.2<br>Locale | AIX 4.1<br>Locale | Country<br>Name         | Country<br>Code | Supported<br>Code Set | Database<br>Code Page | EBCDIC AR<br>CCSID                               | ASCII AR<br>CCSID                |
|-------------------|-------------------|-------------------------|-----------------|-----------------------|-----------------------|--------------------------------------------------|----------------------------------|
| sh_YU             | sh_SP             | Serbia<br>(Latin)       | 038             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| sk_SK             | sk_SK             | Slovakia                | 042             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| si_SI             | si_SI             | Slovenia                | 386             | ISO8859-2             | 912                   | 870                                              | 852 912 1250<br>1282             |
| es_ES             | es_ES             | Spain                   | 034             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Es_ES             | Es_ES             | Spain                   | 034             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| sv_SV             | sv_SE             | Sweden                  | 046             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Sv_SV             | Sv_SE             | Sweden                  | 046             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| fr_CH             | fr_CH             | Switzerland<br>(French) | 041             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Fr_CH             | Fr_CH             | Switzerland<br>(French) | 041             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| de_CH             | de_CH             | Switzerland<br>(German) | 041             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| De_CH             | De_CH             | Switzerland<br>(German) | 041             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| Zh_TW             | Zh_TW             | Taiwan                  | 081             | big5                  | 950                   | 937                                              | 938 948 950<br>964               |
| zh_TW             | zh_TW             | Taiwan                  | 081             | IBM-eucTW             | 964                   | 937                                              | 938 948 950<br>964               |
| th_TH             | th_TH             | Thailand                | 066             | TIS620.2533           | 874                   | 838                                              | 874                              |

Table 17. CCSID Support for DataJoiner for AIX DRDA Application Server (continued)

| AIX 3.2<br>Locale | AIX 4.1<br>Locale | Country<br>Name     | Country<br>Code | Supported<br>Code Set | Database<br>Code Page | EBCDIC AR<br>CCSID                               | ASCII AR<br>CCSID                |
|-------------------|-------------------|---------------------|-----------------|-----------------------|-----------------------|--------------------------------------------------|----------------------------------|
| tr_TR             | tr_TR             | Turkey              | 090             | ISO8859-9             | 920                   | 1026                                             | 857 920 1254<br>1281             |
| en_GB             | en_GB             | U.K.                | 044             | ISO-8859-1            | 819                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| En_GB             | En_GB             | U.K.                | 044             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |
| C                 | C                 | U.S.A.<br>(English) | 001             | IBM-850               | 850                   | 037 273 277<br>278 280 284<br>285 297 500<br>871 | 437 819 850<br>1051 1252<br>1275 |

a: There is no EBCDIC AR CCSID support for 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 encoded on code page 420 to code page 1089 would result in some characters being lost.

## 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 18. 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         |
| 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 AIX."

**Note:** All the units in this table are in KBs.

| 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>  | Agent stack size                                     | Not applicable for DataJoiner for AIX |
| <i>agentpri</i>        | Priority of Agents                                   | -1 (system) [ -1, 41 - 125 ]          |
| <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               | Null [ Null, TCPIP, APPC ]            |
| <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                                 | home directory of instance owner [ any valid path ] |
| <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                          | Null [ any valid string ]                           |
| <i>dir_path_name</i>   | Directory path name in DCE namespace                  | ./:subsys/database [ any valid path ]               |
| <i>dir_type</i>        | Directory services type                               | NONE [ NONE, DCE ]                                  |
| <i>dos_rqrioblk</i>    | DOS requester I/O block size                          | Not applicable for DataJoiner for AIX               |
| <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                                | restart [ access, restart ]                         |
| <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 ]                |
| <i>maxdari</i>         | Maximum number of DARI processes                      | -1 (maxagents) [ -1, 0 - maxagents ]                |
| <i>maxtotfilop</i>     | Maximum total files open per application              | Not applicable for DataJoiner for AIX               |
| <i>min_priv_mem</i>    | Minimum committed private memory                      | Not applicable for DataJoiner for AIX               |
| <i>mon_heap_sz</i>     | Database system monitor heap size                     | 48 [ 0 - 60000 ]                                    |
| <i>nname</i>           | NetBIOS node name                                     | Not applicable for DataJoiner for AIX               |
| <i>numbd</i>           | Maximum number of concurrently active databases       | 8 [ 1 - 60000 ]                                     |
| <i>objectname</i>      | IPX/SPX database manager object name                  | Null [ any valid string ]                           |
| <i>priv_mem_thresh</i> | Private memory threshold                              | Not applicable for DataJoiner for AIX               |
| <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                                | Null [ any valid string ]                    |
| <i>rqrioblk</i>       | Client I/O block size                                          | 32767 [ 4096 - 65535 ]                       |
| <i>sheapthres</i>     | Sort heap threshold                                            | 4096 [ 250 - 524288 ]                        |
| <i>sqlstmtsz</i>      | SQL statement size                                             | 256 [ 0 - 32767 ]                            |
| <i>svcename</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                            | Primary group of instance owner              |
| <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 ]                    |
| <i>tp_mon_name</i>    | Transaction processor monitor name (for all UNIX environments) | Not applicable for DataJoiner for Windows NT |
| <i>tpname</i>         | APPC transaction program name                                  | Null [ CICS, ENCINA, blank, another 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 256K, the agent stack can grow to a maximum of 256K.
- If the agent stack size is greater than 256K, the maximum agent stack size is rounded up to the nearest 1M 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 has been installed. The default values and valid ranges for DB2 for Windows NT are given. Configuration parameters that are not used on this platform are indicated by the string: "Not applicable for DB2 for Windows NT". 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.

| 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                      | 1000 [ 2* <i>maxappls</i> - 524288 ] |

|                        |                                                       |                                                |
|------------------------|-------------------------------------------------------|------------------------------------------------|
| <i>catalogcache_sz</i> | Catalog Cache Size                                    | 64 [ 1 - <i>dbheap</i> ]                       |
| <i>chnpggs_thresh</i>  | Changes pages threshold                               | 60 [ 5 - 80 ]                                  |
| <i>copyprotect</i>     | Copy Protection Enable                                | Not applicable for DataJoiner for AIX          |
| <i>dbheap</i>          | Database Heap                                         | 1200 [ 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>dchktime</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                        | 100 [ 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                                     | 1000 [ 4 - 16384 ]                             |
| <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                 | 40 [ 1 - 1000 ]                                |
| <i>maxfilop</i>        | Maximum number of database files open per application | 64 [ 2 - 1950 ]                                |
| <i>maxlocks</i>        | Maximum percent of lock list before escalation        | 10 [ 1 - 100 ]                                 |
| <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. National Language Support

This appendix contains information about the National Language Support (NLS) provided by DataJoiner, including information about countries, languages, and code pages (code sets) supported and how to configure and use DataJoiner NLS features in both your applications and databases.

**Note:** The replication administration tool is not DBCS-enabled.

---

### Setting Up Your Language Environment

All language specific messages, help panels, and application defaults are stored independently of the programs, and are translated in the following languages and encoded in the following codesets:

|                           |           |         |
|---------------------------|-----------|---------|
| German in Germany         | IBM-850   | (De_DE) |
| German in Germany         | ISO8859-1 | (de_DE) |
| English in United States  | IBM-850   | (En_US) |
| Spanish in Spain          | IBM-850   | (Es_ES) |
| Spanish in Spain          | ISO8859-1 | (es_ES) |
| French in France          | IBM-850   | (Fr_FR) |
| French in France          | ISO8859-1 | (fr_FR) |
| Italian in Italy          | IBM-850   | (It_IT) |
| Italian in Italy          | ISO8859-1 | (it_IT) |
| Japanese in Japan         | IBM-932   | (Ja_JP) |
| Japanese in Japan         | IBM-eucJP | (ja_JP) |
| Korean in Korea           | IBM-eucKR | (ko_KR) |
| Portuguese in Brazil      | ISO8859-1 | (pt_BR) |
| Simplified-Chinese in PRC | IBM-eucCN | (zh_CN) |

You specify the language environment you wish to use by installing the message option for the desired language, then setting the **LANG** environment variable to the desired locale. For example if you want to use Fr\_FR messages, you must set **LANG** to Fr\_FR and have the Fr\_FR message option installed.

The selected message catalog options as well as the messages for the installation scripts are placed in the /usr/lpp/djx\_02\_01\_01/msg/%L directories on the target workstation where %L is equal to the locale name of the message catalog. Selected DBA Utility messages and resources are placed in the following directories:

```
| /usr/lpp/djx_02_01_01/msg/%L
| /usr/lpp/djx_02_01_01/dba/res/%L
```

```
|
| When the djx_02_01_01.client option is installed, a link is made for the
| /usr/lpp/djx_02_01_01/msg/En_US directory to /usr/lpp/djx_02_01_01/msg/prime.
```

```
|
| Similarly, when the djx_02_01_01.dd option is installed, a link is made for the
| /usr/lpp/djx_02_01_01/dba/res/En_US directory to
| /usr/lpp/djx_02_01_01/dba/res/prime.
```

The prime or default directory is the directory that is searched for messages if the user's locale does not match any of the installed messages. For example, if the user's locale is set to No\_NO (which does not have translated messages) the messages in prime (the default is En\_US) will be used instead. The prime directory can be changed by root simply by changing the links. For example, if users are all working in IBM-850 French locales such as Fr\_BE, Fr\_CA, Fr\_CH and Fr\_FR, it might be preferable to install the French (Fr\_FR) messages and make a link for the Fr\_FR message directory to prime as follows:

```
ln -s -f /usr/lpp/djx_02_01_01/msg/Fr_FR \ /usr/lpp/djx_02_01_01/msg/prime
ln -s -f /usr/lpp/djx_02_01_01/dba/res/Fr_FR \ /usr/lpp/djx_02_01_01/dba/res/prime
```

Note that if prime is changed it will affect every instance for this version of the product.

If more than one "default" directory is desired, the following methods can be used:

- Create a link for an existing /usr/lpp/djx\_02\_01\_01/msg/%L directory to /usr/lpp/djx\_02\_01\_01/msg/NewLocale where NewLocale is the name of a new locale such as Fr\_BE.

or

- Create a new directory /usr/lpp/djx\_02\_01\_01/msg/newLocale and either copy the files from an existing /usr/lpp/djx\_02\_01\_01/msg/%L directory or create links for the files in the existing /usr/lpp/djx\_02\_01\_01/msg/%L directory to the /usr/lpp/djx\_02\_01\_01/msg/newLocale directory.

For example, suppose the installer wants to support the En\_US, No\_NO, Po\_PO, Fr\_BE, Fr\_CA, Fr\_CH, Fr\_FR, fr\_BE, fr\_CA, fr\_FR, and fr\_CH locales messages with the following default mappings:

- No\_NO and Po\_PO to En\_US
- Fr\_BE, Fr\_CA, and Fr\_CH to Fr\_FR
- fr\_BE, fr\_CA, and fr\_CH to fr\_FR.

This can be accomplished as follows:

- Install the djx\_02\_01\_01.client, the djx\_02\_01\_01.msg.Fr\_FR.client, and the djx\_02\_01\_01.fr\_FR.client options

This will create the directories:

```
/usr/lpp/djx_02_01_01/msg/En_US
/usr/lpp/djx_02_01_01/msg/Fr_FR
/usr/lpp/djx_02_01_01/msg/fr_FR
```

and the link for:

```
/usr/lpp/djx_02_01_01/msg/En_US to /usr/lpp/djx_02_01_01/msg/prime
```

This link establishes the default prime as En\_US, and thus No\_NO and Po\_PO will default to En\_US.

- Create links for /usr/lpp/djx\_02\_01\_01/msg/Fr\_FR to:
  - /usr/lpp/djx\_02\_01\_01/msg/Fr\_BE

- |                   – /usr/lpp/djx\_02\_01\_01/msg/Fr\_CA
- |                   – /usr/lpp/djx\_02\_01\_01/msg/Fr\_CH
- |                   • Create links for /usr/lpp/djx\_02\_01\_01/msg/fr\_FR to:
- |                   – /usr/lpp/djx\_02\_01\_01/msg/fr\_BE
- |                   – /usr/lpp/djx\_02\_01\_01/msg/fr\_CA
- |                   – /usr/lpp/djx\_02\_01\_01/msg/fr\_CH

**Note:** If you are in Australia or Latin America, you may wish to set your locale to En\_US (IBM-850) or en\_US (ISO8859-1) before database manager installation to match code set requirements and the language of the messages to be selected at installation.

If you are in Latin America, you can also choose to set the 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.

---

## Setting Up Your Application and Database NLS Environments

DataJoiner NLS support is based on specific code set and territory settings that are established when a database is created or when an application executes. The code set value is mapped to an IBM-defined code page, and the territory is mapped to an IBM-defined country code.

In DataJoiner terminology, the code set is a character string identifier for the code page, and the territory is a character string identifier for the country code. A code page is a mapping of 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 set "IBM-850," code page 850 represents the character "A" as hexadecimal 41. The country code setting establishes any national conventions that are supported by the database manager. Each country is also represented by a numeric identifier. For example, territory US, country code 001 represents the United States.

This information is used by the database managers as follows:

### 1. Database Creation

During database creation, if the territory and codeset are not specified on the CREATE DATABASE command (or API), DataJoiner uses the territory and codeset of the application that invokes the command.

**Note:** In AIX the territory value is the second half of a locale which is a combination of language and territory (*language\_territory*), for example: En\_US; specify only the territory value on the create database command.

This defaulting approach adheres to portability guidelines for designing international applications and products that by their nature, are required to work in different countries. These guidelines call for dynamically providing the NLS functions that are required by the active environment.

When you create databases, you need to either provide proper territory and codeset specifications or ensure that the operating environment is set to meet your country's application requirements.

For internal processing, DataJoiner maps the territory value to a country code and the codeset to an IBM-defined code page value. (See Table 20 on page 338 for a mapping of territory and codeset to country code and code page.) In turn, the country code and code page are used to determine the default collating sequence that will be used for the database. This default collating sequence may be explicitly overridden, see "CREATE DATABASE" in the *DB2 Command Reference* for information on how to override the collating sequence.

All this NLS information, including the collating sequence, is stored in the database configuration file. Once created this information cannot be changed since it is used to determine additional NLS processing. For example, the collating sequence is used to generate and process indexes that are required on character data type columns. It is also used by character string comparison functions, and by the SORT BY, and GROUP BY clauses in SQL statements.

## 2. Database Access

When the database is accessed, the derived country code of the application determines the date and time presentation (display and print) formats. DataJoiner transforms the date and time from the internal storage format to the appropriate country-external format. For Extended Services for OS/2 and DB2 for OS/2 database clients that do not pass their country code to the database server, the database country code is used.

Also, while accessing the database, the application code page is compared to the code page of the database to determine if character data conversion is required. See "Conversion Between Different Code Pages" on page 333 for more information about character conversion.

---

## 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. 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 20 on page 338 for a complete list of supported locales, code page equivalents.

## **Deriving Locales in Application Programs**

Locales are specific to UNIX-based operating systems. There are two locales:

- The environment locale allows you to specify the language, currency symbol, and so on, that you want to use.
- The program locale contains the current language, currency symbol, and so on, of a program that is executing.



When your program is started, it gets a default C locale. It does **not** get a copy of the environment locale. Your program has a few choices:

- Ignore the environment locale. Your program could hardcode some options. For example, your program could set the language to *spanish* with the `setlocale()` function.
- Copy the environment locale to the program locale.
- Ignore the environment locale. Use whatever defaults you get from the operating system.

## How DataJoiner Derives Locales

With UNIX, the active locale that is used by DataJoiner is determined from the LC-CTYPE portion of the locale. For details, see the NLS documentation for your operating system.

- If LC\_CTYPE of the program locale has a value other than that of 'C', DataJoiner will use this value to determine the application code page by mapping it to its corresponding code page.
- If LC\_CTYPE has the value of 'C' (the 'C' locale), DataJoiner will set the program locale according to the environment locale using the `setlocale()` function.
- If LC\_CTYPE still has a value of 'C', DataJoiner will use its default locale for that platform. See the *DB2 SDK Building Your Applications* book for your platform for information on the default locale for that platform.
- If LC\_CTYPE's value is no longer 'C', its new value will be used to map to a corresponding code page.

## Programming Considerations

It is strongly recommended that applications be precompiled, bound, compiled, and executed using the same code page. This is because data conversions by the server can occur in both the bind and the execution phases. Users should ensure that the same conversion tables are used by binding and executing with the same active code page. For a discussion of how applications determine the active code page, see "Deriving Code Page Values" on page 328.

Any external data that is obtained by the application will be assumed to be in the application code page. This includes data that is obtained from a file or from user input. Make sure that data sources outside the application uses the same code page as the application.

If you use host variables that use graphic data in your C or C++ applications, there are special precompiler, application performance, and application design issues you need to consider. For a detailed discussion of these considerations, or if you deal with EUC code sets in your applications, see the *DB2 Application Programming Guide*.

## Coding SQL Statements

The coding of SQL statements is not language dependent. The SQL keywords must be typed as they are shown in this book, although they may be typed in uppercase, lowercase, or mixed case. The names of database objects, host variables, and program labels that occur in an SQL statement cannot contain characters outside the extended character set that is supported by your code page. For more information about extended character sets, see the *DB2 SQL Reference*.

Constant character strings in static SQL statements are converted at bind time from the application code page to the database code page, and will be used at execution time in this database code page representation. To avoid such conversions if they are not desired, you can use host variables in place of string constants.

The server does not convert file names. To code a file name, either use the ASCII invariant set, or provide the path name in hexadecimal values that are physically stored in the file system.

## Coding Remote Stored Procedures

When coding stored procedures that will be running remotely, the following considerations apply:

- Data in a stored procedure must be in the database code page.
- Data passed to or from a stored procedure using an SQLDA with a character data type must really contain character data. Numeric data and data structures must never be passed if the client application code page is different from the database code page. This is because the server will convert all character data in an SQLDA. To avoid character conversion, you can pass data by defining it in binary string format by using a data type of BLOB or by defining the character data as FOR BIT DATA.

## Precompiling and Binding

At precompile/bind time, the precompiler is the executing application. The active code page when the database connection was made prior to the precompile request is used for precompiled statements and for any character data returned in the SQLCA.

## Executing an Application

At execution time, the active code page of the user application when a database connection is made is in effect for the duration of the connection. All data is interpreted based on this code page; this includes dynamic SQL statements, user input data, user output data, and character fields in the SQLCA.

## A Note of Caution

Failure to follow these guidelines may produce unpredictable results. These conditions cannot be detected by the database manager, so no error or warning message will

result. For example, a C application contains the following SQL statements operating against a table T1 with one column that is defined as C1 CHAR(20):

```
(0) EXEC SQL CONNECT TO GLOBALDB;
(1) EXEC SQL INSERT INTO T1 VALUES ('a-constant');
 strcpy(sqlstmt, "SELECT C1 FROM T1 WHERE C1='a-constant');
(2) EXEC SQL PREPARE S1 FROM :sqlstmt;
```

Where:

```
application code page at bind time = x
application code page at execution time = y
database code page = z
```

At bind time 'a-constant' in statement (1) is converted from code page x to code page z. This conversion can be noted as (x to z).

At execution time, 'a-constant' (x to z) is inserted into the table when statement (1) is executed. However, the WHERE clause of statement (2) will be executed with 'a-constant' (y to z). If the codepoints in the constant are such that the two conversions (x to z and y to z) yield different results, the SELECT in statement (2) will fail to retrieve the data that is inserted by statement (1).

This appendix contains information about the National Language Support (NLS) provided by DataJoiner, including information about countries, languages, and code pages (code sets) supported and how to configure and use DataJoiner NLS features in both your applications and databases.

---

## Conversion Between Different Code Pages

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 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.

This database conversion will occur on the database server machine for both conversions from the application code page to the database code page and from the database code page to the application code page.

You can minimize or eliminate client/server character conversion in some situations. For example, you could create a DB2 for AIX database that uses code page 850 to match an OS/2 and DOS client application environment that predominately uses code page 850.

**Note:** The DB2 for OS/2 Version 1.0 or Version 1.2 database server does not support character conversion between different code pages. Ensure that the code pages on server and client are compatible. See Table 19 on page 336 for the code page conversions that are supported.

- When a client or application 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 AIX 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 for AIX machine from DB2 for MVS is converted by DDCS for AIX. 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.<sup>11</sup> 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.
- A DB2 product or platform that does not support, or that does not have support installed, for the desired combination of code pages. In this case, an SQLCODE -322 (SQLSTATE 57017) is returned when you try to run your application.

## Character Substitutions During Conversions

When your application converts from one code page to another, it is possible that one or more characters are not represented in the target code page. If this occurs, DataJoiner inserts a *substitution* character into the target string in place of the character that has no representation. The replacement character is then considered a valid part of the string. In situations where a substitution occurs, the SQLWARN10 indicator in the SQLCA is set to 'W'.

---

11. However, a literal that is inserted into a column that is defined as FOR BIT DATA could be converted if that literal was part of an SQL statement which was converted.

**Note:** Any character conversions that result from using the WCHARTYPE CONVERT precompiler option will not flag a warning if any substitutions take place.

## 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 19 to avoid a possible loss of character data.

- If the data is derived from operations 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 19 on page 336 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.

**Note:** 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 19. Supported Code Page Conversions

| Code Pages                                               | Countries                                                                                                                                                                         |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 437, 819, 850, 863, 1004, 1051, 1252 <sup>a</sup> , 1275 | Austria, Australia, Belgium, Brazil, Canada, Denmark, Finland, France, Germany, Iceland, Italy, Latin America, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, UK, USA |
| 813, 869, 1253 <sup>a</sup> , 1280                       | Greece                                                                                                                                                                            |
| 852, 912, 1250 <sup>a</sup> , 1282                       | Croatia, Czech Republic, Hungary, Poland, Romania, Serbia/Montenegro (Latin), Slovakia, Slovenia                                                                                  |
| 855, 915, 1251 <sup>a</sup> , 1283                       | Bulgaria, FYR Macedonia, Serbia/Montenegro (Cyrillic)                                                                                                                             |
| 857, 920, 1254 <sup>a</sup> , 1281                       | Turkey                                                                                                                                                                            |
| 862, 916, 1255 <sup>a</sup>                              | Israel                                                                                                                                                                            |
| 864, 1046, 1089, 1256 <sup>a</sup>                       | Arabic countries                                                                                                                                                                  |
| 866, 915, 1251 <sup>a</sup> , 1283                       | Russia                                                                                                                                                                            |
| 932, 942 <sup>b</sup> , 943, 954                         | Japan <sup>e</sup>                                                                                                                                                                |
| 938, 948 <sup>c</sup> , 950, 964                         | Taiwan <sup>e</sup>                                                                                                                                                               |
| 949, 970 <sup>d</sup>                                    | Korea <sup>e</sup>                                                                                                                                                                |
| 1381, 1383 <sup>d</sup>                                  | People's Republic of China <sup>e</sup>                                                                                                                                           |

**Note:**

- a** Code pages 1250, 1251, 1252, 1253, 1254, 1255, and 1256 are treated as server code pages by the Windows NT database manager.
- b** Code page 932 and 942 are treated as equivalent by DataJoiner, so no character conversion is performed between them.
- c** Code page 938 and 948 are treated as equivalent by DataJoiner, so no character conversion is performed between them.
- d** 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.
- e** Conversion support is optionally installable for these countries.

## Character Conversion Expansion Factor

When your application successfully completes an attempt to connect to a DataJoiner database server, you should consider the following fields in the returned SQLCA:

- The second token in the SQLERRMC field (tokens are separated by X'FF') indicates the code page of the database. The ninth token in the SQLERRMC field indicates the code page of the application. Querying the application's code page and comparing it to the database's code page informs the application whether it has established a connection that will undergo character conversions.
- The first and second entries in the SQLERRD array. SQLERR(1) contains an integer value equal to the maximum expected expansion or contraction factor for the length

of mixed character data (CHAR data types) when converted to the database code page from the application code page. SQLERRD(2) contains an integer value equal to the maximum expected expansion or contraction factor for the length of mixed character data (CHAR data types) when converted to the application code page from the database code page. A value of 0 or 1 indicates no expansion; a value greater than 1 indicates a possible expansion in length; a negative value indicates a possible contraction. Refer to the *DB2 SQL Reference* for details on using the CONNECT statement.

The considerations for graphic string data should not be a factor in unequal code page situations. Each string always has the same number of characters, regardless of whether the data is in the application or the database code page.

For more information on dealing with unequal code page situations, see the *DB2 Application Programming Guide*.

---

## Locale, Code Page, and Host CCSID Support

Table 20 on page 338 shows the locales, code sets, code pages, and CCSIDs that are supported by DataJoiner. Each row of this table shows the supported conversions between the database code page and the application code page (code set). These are the code pages under which the database and application must be running for client/server connectivity to take place.

The following list explains each column of the table:

1. **Supported Locale** lists the locales that are supported by DataJoiner. The locale (territory) is mapped to the country code.
2. **Country Code** shows the country code that maps to the territory of the supported locale.
3. **Supported Code Set** shows the AIX code set associated with the supported locale. The code set is mapped to the Database Code Page.
4. **Database Code Page** shows the IBM-defined code page under which the database is created.
5. **AIX Application Code Page** shows the IBM-defined code page that an AIX application can use when connecting to a DataJoiner database.
6. **OS/2 Application Code Page** shows the IBM-defined code page that an OS/2 client application can use when connecting to a DataJoiner database.
7. **DOS Application Code Page** shows the IBM-defined code page that a DOS client application can use when connecting to a DataJoiner database.
8. **Windows Application Code Page** shows the IBM-defined code page that a Windows client application can use when connecting to a DataJoiner database.
9. **Host Supported CCSID** shows the IBM-defined Coded Character Set Identifier (CCSID) representing the encoding of data coming from the host that is supported by DataJoiner.

**Note:** Check the manuals for your host database product to determine which CCSIDs are supported in your host environment.

10. **Country Name** shows the name of the country.

Table 20. Supported Locales, Code Sets, Code Pages, and CCSIDs

| Supported locale | Country code | Supported code set | Database code page | AIX application code page | OS/2 application code page | DOS application code page | Windows application code page | Host supported CCSID | Country name          |
|------------------|--------------|--------------------|--------------------|---------------------------|----------------------------|---------------------------|-------------------------------|----------------------|-----------------------|
| ar_AR            | 785          | ISO8859-6          | 1089               | 1089                      | 864                        | 864                       | -                             | Note 5               | Arabic                |
| Ar_AR            | 785          | IBM-1046           | 1046               | 1046                      | 864                        | 864                       | -                             | 420 (Note 5)         | Arabic                |
| C (Note 1)       | 001          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | Australia             |
| C (Note 1)       | 001          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 284                  | Latin America         |
| C (Note 1)       | 001          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | United States         |
| cs_CS            | 042          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Czech                 |
| da_DK            | 045          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 277                  | Denmark               |
| Da_DK            | 045          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 277                  | Denmark               |
| de_CH            | 041          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Switzerland (German)  |
| De_CH            | 041          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Switzerland (German)  |
| de_DE            | 049          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 273                  | Germany               |
| De_DE            | 049          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 273                  | Germany               |
| el_GR            | 030          | ISO8859-7          | 813                | 813                       | -                          | -                         | -                             | 875                  | Greece                |
| en_GB            | 044          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 285                  | United Kingdom        |
| En_GB            | 044          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 285                  | United Kingdom        |
| en_US            | 001          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037, 500 (Note 3)    | United States, Canada |
| En_US            | 001          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037, 500 (Note 3)    | United States, Canada |
| es_ES            | 034          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 284                  | Spain                 |
| Es_ES            | 034          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 284                  | Spain                 |
| fi_FI            | 358          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 278                  | Finland               |
| Fi_FI            | 358          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 278                  | Finland               |
| fr_BE            | 032          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Belgium (French)      |
| Fr_BE            | 032          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Belgium (French)      |



Table 20. Supported Locales, Code Sets, Code Pages, and CCSIDs (continued)

| Supported locale | Country code | Supported code set | Database code page | AIX application code page | OS/2 application code page | DOS application code page | Windows application code page | Host supported CCSID | Country name         |
|------------------|--------------|--------------------|--------------------|---------------------------|----------------------------|---------------------------|-------------------------------|----------------------|----------------------|
| fr_CA            | 002          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037, 500 (Note 3)    | Canada (French)      |
| Fr_CA            | 002          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037, 500 (Note 3)    | Canada (French)      |
| fr_CH            | 041          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Switzerland (French) |
| Fr_CH            | 041          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Switzerland (French) |
| fr_FR            | 033          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 297                  | France               |
| Fr_FR            | 033          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 297                  | France               |
| is_IS            | 354          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 871                  | Iceland              |
| hr_HR            | 038          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Croatia              |
| hu_HU            | 036          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Hungary              |
| Is_IS            | 354          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 871                  | Iceland              |
| it_IT            | 039          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 280                  | Italy                |
| It_IT            | 039          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 280                  | Italy                |
| iw (Note 4)      | 972          | ISO8859-8          | 916                | 916                       | 862                        | 862                       | -                             | 424                  | Israel               |
| Ja_JP            | 081          | IBM-932            | 932                | 932                       | 942, 932                   | 932                       | 932                           | 930, 939, 5026, 5035 | Japan (Note 6)       |
| ko_KR            | 088          | IBM_euckR          | 970                | 970                       | 934, 944                   | -                         | -                             | -                    | Korea                |
| mk_MK            | 389          | ISO8859-5          | 915                | 915                       | 855                        | 855                       | -                             | 1025                 | FYR Macedonia        |
| nl_BE            | 032          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Belgium (Dutch)      |
| NI_BE            | 032          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 500                  | Belgium (Dutch)      |
| nl_NL            | 031          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | Netherlands          |
| NI_NL            | 031          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | Netherlands          |
| no_NO            | 047          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 277                  | Norway               |
| No_NO            | 047          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 277                  | Norway               |
| pl_PL            | 048          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Poland               |
| pt_PT            | 351          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | Portugal             |
| Pt_PT            | 351          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 037                  | Portugal             |

Table 20. Supported Locales, Code Sets, Code Pages, and CCSIDs (continued)

| Supported locale | Country code | Supported code set | Database code page | AIX application code page | OS/2 application code page | DOS application code page | Windows application code page | Host supported CCSID | Country name                 |
|------------------|--------------|--------------------|--------------------|---------------------------|----------------------------|---------------------------|-------------------------------|----------------------|------------------------------|
| ro_RO            | 038          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Romania                      |
| ru_RU            | 007          | ISO8859-5          | 915                | 915                       | 866                        | 866                       | -                             | 1025                 | Russia                       |
| sh_YU            | 038          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Serbia (Latin)               |
| si_SI            | 038          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Slovenia                     |
| sk_CS            | 042          | ISO8859-2          | 912                | 912                       | 852                        | 852                       | -                             | 870                  | Slovakia                     |
| sp_YU            | 381          | ISO8859-5          | 915                | 915                       | 855                        | 855                       | -                             | 1025                 | Serbia/Montenegro (Cyrillic) |
| sv_SV            | 046          | ISO8859-1          | 819                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 278                  | Sweden                       |
| Sv_SV            | 046          | IBM-850            | 850                | 819, 850                  | 850, 437                   | 850, 437                  | Note 2                        | 278                  | Sweden                       |
| th_TH            | 066          | TIS620.2533        | 874                | 874                       | -                          | -                         | -                             | Note 7               | Thailand                     |
| tr_TR            | 090          | ISO8859-9          | 920                | 920                       | 857                        | 857                       | -                             | 1026                 | Turkey                       |
| zh_CN            | 086          | IBM_eucCN          | 1383               | 936, 946                  | -                          | -                         | -                             | -                    | PRC                          |
| zh_TW            | 088          | big5               | 950                | 950                       | 938, 948                   | -                         | -                             | -                    | Big-5                        |

**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 running 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.

## DBCS Code Page Support

Combined SBCS/DBCS code pages contain both single- and double-byte code points, and thus imply both an SBCS and DBCS code page. Table 21 on page 342 summarizes the relationships between the supported code pages (which is a combined SBCS/DBCS code page), and the SBCS and DBCS code pages implied by this combined code page.

Table 21. Relationships Between SBCS and DBCS Code Pages

| Supported Code Page                                                                                                                                                                                                                                                                                                    | Platform                        | Implied SBCS Code Page | Implied DBCS Code Page | Name                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|------------------------|------------------------|------------------------|
| 932                                                                                                                                                                                                                                                                                                                    | OS/2, AIX                       | 897                    | 301                    | Japanese               |
| 942                                                                                                                                                                                                                                                                                                                    | OS/2                            | 1041                   | 301                    | Japanese Extended      |
| 943                                                                                                                                                                                                                                                                                                                    | Windows, Windows NT, Windows 95 | 1041                   | 941                    | Japanese               |
| 938                                                                                                                                                                                                                                                                                                                    | OS/2                            | 904                    | 927                    | Traditional Chinese    |
| 948                                                                                                                                                                                                                                                                                                                    | OS/2                            | 1043                   | 927                    | Trad. Chinese Extended |
| 949                                                                                                                                                                                                                                                                                                                    | OS/2                            | 1088                   | 951                    | Korean                 |
| 950                                                                                                                                                                                                                                                                                                                    | OS/2 AIX HP                     | 1114                   | 947                    | Trad. Chinese big5     |
| 1381                                                                                                                                                                                                                                                                                                                   | OS/2                            | 1115                   | 1380                   | Simplified Chinese     |
| Notes:                                                                                                                                                                                                                                                                                                                 |                                 |                        |                        |                        |
| <ul style="list-style-type: none"> <li>• Code pages 942 and 948 are only supported for database clients running under OS/2.</li> <li>• Code pages 932 and 942 are treated as identical, as are code pages 938 and 948.</li> <li>• Platform indicates the operating system where the code page is supported.</li> </ul> |                                 |                        |                        |                        |

## EUC Code Page Support

Extended UNIX Code (EUC) code pages are actually a combination of up to four character groups, known as G0, G1, G2, and G3. Each group can be identified with a code page in addition to the combined EUC code page identifier. The table that follows summarizes the relationships between the supported code pages (which are EUC code pages) and the G0 through G3 code pages that are implied by the EUC code page.

Table 22. Relationships Between EUC Code Pages and Character Groups

| Supported Code Page | Platform            | Implied Code Pages |      |      |     | Name                    |
|---------------------|---------------------|--------------------|------|------|-----|-------------------------|
|                     |                     | G0                 | G1   | G2   | G3  |                         |
| 954                 | AIX, HP-UX, Solaris | 895                | 952  | 4992 | 953 | Japanese EUC            |
| 964                 | AIX, HP-UX, Solaris | 367                | 960  | 961  | n/a | Traditional Chinese EUC |
| 970                 | AIX, HP-UX, Solaris | 367                | 971  | n/a  | n/a | Korean EUC              |
| 1383                | AIX, Solaris        | 367                | 1382 | n/a  | n/a | Simplified Chinese EUC  |

Database and client application support for graphic (pure double-byte character) data while running under EUC code pages with character encodings that are greater than two bytes in length is limited. DataJoiner has strict rules for graphics data that require all data to be exactly two bytes wide. These restrictions essentially rule out many characters from both the Japanese and Traditional Chinese EUC code pages.

To overcome this restriction, support is provided at both the application level and the database level to represent Japanese and Traditional Chinese EUC graphic data that uses another coding scheme. A database that is created under Japanese and Traditional Chinese EUC code pages will store and manipulate graphic data by using the UCS-2 code set. Similarly, applications that run under those code pages will send graphic data to the database server as UCS-2-encoded data.

This support enables applications that run under EUC code pages to access the same type of data as those running under DBCS code pages.

For information regarding application development in EUC environments, refer to the *DB2 Application Programming Guide* and the *DB2 SQL Reference*.

The IBM-defined code page identifier associated with UCS-2-encoded data for DataJoiner 13844. The support for UCS-2-encoded data is at Level 1 of the standard (no combining characters).

---

## DataJoiner Code Page Support

Because DataJoiner deals with multiple data sources, unequal code pages can exist in a heterogeneous database environment. In this situation, DataJoiner identifies the DataJoiner code page to the data source client code. The data source character set transformation logic returns the data to DataJoiner in the DataJoiner code page. If you use an invalid configuration (for example, if you attempt to transfer data from Sybase running the Sybase code page equivalent to 850 to DataJoiner in code page 8859), then the client code must either detect this error or map the data by using their own semantics. DataJoiner relies on non-IBM data sources for translation at the data source.

## Specifying Code Pages

DataJoiner supports single-byte (SBCS) and double-byte (DBCS) character sets. These include the IBM PC code sets and a subset of the ISO code sets that are identified in Table 23.

*Table 23. Supported Code Sets*

| Code Set  | Code Page | SBCS or DBCS               |
|-----------|-----------|----------------------------|
| IBM-850   | 850       | SBCS                       |
| IBM-932   | 932       | Combined SBCS 897/DBCS 301 |
| IBM-1046  | 1046      | SBCS                       |
| ISO8859-1 | 819       | SBCS                       |
| ISO8859-2 | 912       | SBCS                       |
| ISO8859-6 | 1089      | SBCS                       |
| ISO8859-7 | 813       | SBCS                       |
| ISO8859-8 | 916       | SBCS                       |

Table 23. Supported Code Sets (continued)

| Code Set  | Code Page | SBCS or DBCS |
|-----------|-----------|--------------|
| ISO8859-9 | 920       | SBCS         |

Table 24 provides mappings between code pages and Oracle and Sybase options. Your data sources must be configured to correspond to these mappings, or else the client code must be able to detect the mismatch and flag it as an error, or map the data by using its own semantics. For example, Sybase maps all unmappable characters to '?'. See your data source documentation for information about other non-IBM data sources.

Table 24. Code Page Mappings

| Code Page | Oracle                                          | Sybase |
|-----------|-------------------------------------------------|--------|
| 850       | NLS_LANG=American_America.US7ASCII              | cp850  |
| 932       | NLS_LANG=Japanese_Japan.JA16SJIS                | sjis   |
| 1046      | NLS_LANG=Arabic_UnitedArabEmirates.US7ASCII     | n/a    |
| 819       | NLS_LANG=German_Germany.WE8ISO8859P1            | iso_1  |
| 912       | NLS_LANG=German_Germany.EE8ISO8859P2            | iso_2  |
| 1089      | NLS_LANG=Arabic_UnitedArabEmirates.AR8ISO8859P6 | iso_6  |
| 813       | NLS_LANG=Greek_Greece.EL8ISO8859P7              | iso_7  |
| 916       | NLS_LANG=American_America.IW8ISO8859P8          | iso_8  |
| 920       | NLS_LANG=Turkish_Turkey.TR8ISO8859P9            | iso_9  |
| 950       | NLS_LANG=Chinese_Taiwan.ZHT16BIG5               |        |
| 970       | NLS_LANG=Korean_Korea.KO16KSC5601               |        |
| 1383      | NLS_LANG=Chinese_China.ZHS16CGB231280           |        |

For data sources that use the generic ODBC API, DataJoiner administrators must enter the code set value for each data source that is different from the default value in the SYSCAT.SERVERS catalog view. For more information about SYSCAT.SERVERS, see Appendix B, "System Catalog Views", in *DataJoiner Application Programming and SQL Reference Supplement*.

For character data conversion to take place between a DataJoiner and a DRDA data source, conversion tables from the source code page to the target code page must exist at DataJoiner and at the DRDA data source. DataJoiner relies on non-DRDA data sources to provide all code page translations. Data sources using the same code set as DataJoiner require no translation.

If no conversion table can be found from the source code page to the target code page, DataJoiner issues an error message.

Administrators must enter the format of date, time, and timestamp variables for each data source in the SYSCAT.SERVERS catalog view. DataJoiner performs the necessary conversion to support the datetime values across multiple databases.

---

## Character Sets

DataJoiner does not, in general, restrict the character set available to an application except as noted below.

### DBCS Character Sets

Each combined SBCS/DBCS code page allows for both single- and double-byte character code points. This is accomplished by reserving a subset of the 256 available code points of each implied SBCS code page identifier for single-byte characters, with the remainder of the code points either undefined or allocated to the first byte of double-byte code points. These code points are shown in the following table.

*Table 25. Mixed Character Set Code Points*

| Supported Mixed Code Page | Code Points for Single-byte Characters | Code Points for First Byte of Double-Byte Characters |
|---------------------------|----------------------------------------|------------------------------------------------------|
| 932                       | x00-7F, xA1-DF                         | x81-9F, xE0-FC                                       |
| 942, 943                  | x00-80, xA0-DF, xFD-FF                 | x81-9F, xE0-FC                                       |
| 938                       | x00-7E                                 | x81-FC                                               |
| 948                       | x00-80                                 | x81-FC                                               |
| 949                       | x00-7F                                 | x8F-FE                                               |
| 950                       | x00-7E                                 | x81-FE                                               |
| 1381                      | x00-7F                                 | x8C-FE                                               |

Code points that are not assigned to either category above are not defined, and are processed as single-byte undefined code points.

Within each implied DBCS code page, there are 256 code points available as the second byte for each valid first byte. These code points are also partitioned into valid and invalid second byte ranges for the purpose of determining whether a DBCS character is properly formed. Note that in DBCS environments, DataJoiner does not perform validity checking on individual double-byte characters.

### Extended UNIX Code (EUC) Character Sets

Each EUC code page allows for both single-byte and up to three different sets of multi-byte character code points. This is accomplished by reserving a subset of the 256 available code points of each implied SBCS code page identifier for single-byte characters, with the remainder of the code points either undefined, allocated as an element of a multi-byte character, or allocated as a single-shift introducer of a multi-byte character. These code points are shown in the following tables.

Table 26. Japanese EUC Code Points

| Group | 1st Byte | 2nd Byte | 3rd Byte | 4th Byte |
|-------|----------|----------|----------|----------|
| G0    | x20-x7E  | n/a      | n/a      | n/a      |
| G1    | xA1-xFE  | xA1-xFE  | n/a      | n/a      |
| G2    | x8E      | xA1-xFE  | n/a      | n/a      |
| G3    | x8F      | xA1-xFE  | xA1-xFE  | n/a      |

Table 27. Traditional Chinese EUC Code Points

| Group | 1st Byte | 2nd Byte | 3rd Byte | 4th Byte |
|-------|----------|----------|----------|----------|
| G0    | x20-x7E  | n/a      | n/a      | n/a      |
| G1    | xA1-xFE  | xA1-xFE  | n/a      | n/a      |
| G2    | x8E      | xA1-xFE  | xA1-xFE  | xA1-xFE  |
| G3    | n/a      | n/a      | n/a      | n/a      |

Table 28. Korean EUC Code Points

| Group | 1st Byte | 2nd Byte | 3rd Byte | 4th Byte |
|-------|----------|----------|----------|----------|
| G0    | x20-x7E  | n/a      | n/a      | n/a      |
| G1    | xA1-xFE  | xA1-xFE  | n/a      | n/a      |
| G2    | n/a      | n/a      | n/a      | n/a      |
| G3    | n/a      | n/a      | n/a      | n/a      |

Table 29. Simplified Chinese EUC Code Points

| Group | 1st Byte | 2nd Byte | 3rd Byte | 4th Byte |
|-------|----------|----------|----------|----------|
| G0    | x20-x7E  | n/a      | n/a      | n/a      |
| G1    | xA1-xFE  | xA1-xFE  | n/a      | n/a      |
| G2    | n/a      | n/a      | n/a      | n/a      |
| G3    | n/a      | n/a      | n/a      | n/a      |

Code points that are not assigned to either category above are not defined, and are treated as single-byte undefined code points.

## UCS-2 Character Sets

The UCS-2 encoding scheme is based on a straight assignment of a unique two-byte code point for each character. There are no modes or escape characters to specify modified characters or special cases. There are no protected ranges or shift characters, as is the case with DBCS or EUC encoding schemes. All code points are considered significant.



| Supported Code Page                                                                                                                                                                                             | 1st Byte | 2nd Byte |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|
| 13488 <sup>(1)</sup>                                                                                                                                                                                            | x00-xFF  | x00-xFF  |
| <b>Note:</b><br><br>(1) Support for UCS-2 encoded data is provided only as an internal graphic data representation for applications and databases running under Japanese or Traditional-Chinese EUC code pages. |          |          |

## Character Set for Identifiers

The basic character set that may be used in database names consists of the single-byte uppercase and lowercase Latin letters (A...Z, a...z), the Arabic numerals (0...9) and the underscore character (\_). This list of letters is augmented with the three special characters #, @, and \$ to provide compatibility with host database products. However, these special characters should be used with care in an NLS environment because they are not included in the NLS host (EBCDIC) invariant character set.

When naming database objects (such as tables and views), program labels, host variables, cursors, and statements alphabetic characters from the extended character set may also be used. For example, those letters with diacritical marks. The available characters depend on the code page in use and if you are using the database in a multiple code page environment, you must ensure that all code pages support any alphabetic characters you plan on using from the extended character set. See the *DB2 SQL Reference* for a discussion of delimited identifiers which can be used in SQL statements and can contain characters outside the extended character set.

## Extended Character Set Definition for DBCS Identifiers

In DBCS environments, the extended character set consists of all the characters in the basic character set, plus those that are identified as a letter or digit as follows:

- All double-byte characters in each DBCS code page, except the double-byte space, are valid letters.
- The double-byte space is a special character.
- The single-byte characters available in each mixed code page are assigned to various categories as follows:

### Category

#### Valid Code Points within each Mixed Code Page

**Digits** x30-39

**Letters** x23-24, x40-5A, x61-7A, xA6-DF (A6-DF for code pages 932 and 942 only)

### Special Characters

All other valid single-byte character code points

## Coding of SQL Statements

The coding of SQL statements is not language dependent. SQL is a programming language, and like other programming languages such as C, SQL is language invariant. The SQL keywords must be typed as shown in this book, although they may be typed in uppercase, lowercase, or mixed case. The names of database objects, host variables, and program labels that occur in an SQL statement cannot contain characters outside the DataJoiner extended character set as described above.

---

## Collating Sequences

When the collating (or sort) sequence is not specified, DataJoiner selects a sequence that is based on the code page and country code of the database.

The DataJoiner collating sequence follows the order indicated by the "Character ID" column in the tables documented in the *National Language Design Guide: Volume 2, National Language Support Reference Manual*, SE09-8002. This guide is subsequently referred to as the *NLS Reference Manual*.

Other weight considerations stated in the *NLS Reference Manual* are not used by DataJoiner, which duplicates the order that is given by the "Character ID" column using internal tables.

### Notes:

1. In order to provide consistent sort processing in a client/server and distributed database environment, DataJoiner uses a common sort sequence for all Latin-1 countries. As a result, the sort sequence for Spain and Latin American Spanish-speaking countries does not provide special sort considerations for letters "ch" and "ll".
2. Because code points are likely to be assigned to different characters in a country's EUC, PC DBCS, and UCS-2 code sets, you may obtain different results when you sort the same characters. For more information concerning collation within the database, please refer to the *DB2 SQL Reference Manual*.

For more information about the default collating sequences, see the *NLS Reference Manual*.

---

## Datetime Values

The datetime data types are described below. Although datetime values can be used in certain arithmetic and string operations and are compatible with certain strings, they are neither strings nor numbers.

**Date** A *date* is a three-part value (year, month, and day). The range of the year part is 0001 to 9999. The range of the month part is 1 to 12. The range of the day part is 1 to *x*, where *x* depends on the month.

The internal representation of a date is a string of 4 bytes. Each byte consists of 2 packed decimal digits. The first 2 bytes represent the year, the third byte the month, and the last byte the day.

The length of a DATE column, as described in the SQLDA, is 10 bytes, which is the appropriate length for a character string representation of the value.

**Time** A *time* is a three-part value (hour, minute, and second) that designates a time of day under a 24-hour clock. The range of the hour part is 0 to 24; while the range of the other parts is 0 to 59. If the hour is 24, the minute and second specifications will be zero.

The internal representation of a time is a string of 3 bytes. Each byte is 2 packed decimal digits. The first byte represents the hour, the second byte the minute, and the last byte the second.

The length of a TIME column, as described in the SQLDA, is 8 bytes, which is the appropriate length for a character string representation of the value.

### **Timestamp**

A *timestamp* is a seven-part value (year, month, day, hour, minute, second, and microsecond) that designates a datetime as defined above, except that the time includes a fractional specification of microseconds.

The internal representation of a timestamp is a string of 10 bytes, each of which consists of 2 packed decimal digits. The first 4 bytes represent the date, the next 3 bytes the time, and the last 3 bytes the microseconds.

The length of a TIMESTAMP column, as described in the SQLDA, is 26 bytes, which is the appropriate length for the character string representation of the value.

### **String Representations of Datetime Values**

Values whose data types are DATE, TIME, or TIMESTAMP are represented in an internal form that is transparent to the SQL user. Dates, times, and timestamps can, however, also be represented by character strings, and these representations directly concern the SQL user since there are no constants or variables whose data types are DATE, TIME, or TIMESTAMP. Thus, to be retrieved, a datetime value must be assigned to a character string variable. The character string representation is normally the default format of datetime values that are associated with the country code of the database, unless overridden by specification of the *F* format option when the program is precompiled or bound to the database. See Table 32 on page 352 for a listing of the string formats for the various country codes.

When a valid string representation of a datetime value is used in an operation with an internal datetime value, the string representation is converted to the internal form of the date, time, or timestamp before the operation is performed. The following sections define the valid string representations of datetime values.

### **Date Strings**

A string representation of a date is a character string that starts with a digit

and has a length of at least 8 characters. Trailing blanks may be included; leading zeros may be omitted from the month and day portions.

Valid string formats for dates are listed in Table 1. Each format is identified by name and includes an associated abbreviation and an example of its use.

*Table 30. Formats for String Representations of Dates*

| Format Name                                | Abbreviation | Date Format                      | Example    |
|--------------------------------------------|--------------|----------------------------------|------------|
| International Standards Organization       | ISO          | yyyy-mm-dd                       | 1991-10-27 |
| IBM USA standard                           | USA          | mm/dd/yyyy                       | 10/27/1991 |
| IBM European standard                      | EUR          | dd.mm.yyyy                       | 27.10.1991 |
| Japanese Industrial Standard Christian era | JIS          | yyyy-mm-dd                       | 1991-10-27 |
| Site-defined (Local)                       | LOC          | Depends on database country code | —          |

### Time Strings

A string representation of a time is a character string that starts with a digit and has a length of at least 4 characters. Trailing blanks may be included; a leading zero may be omitted from the hour part of the time and seconds may be omitted entirely. If you choose to omit seconds, an implicit specification of 0 seconds is assumed. Thus, 13.30 is equivalent to 13.30.00.

Valid string formats for times are listed in Table 31. Each format is identified by name and includes an associated abbreviation and an example of its use.

*Table 31. Formats for String Representations of Times*

| Format Name                                | Abbreviation | Time Format                      | Example  |
|--------------------------------------------|--------------|----------------------------------|----------|
| International Standards Organization       | ISO          | hh.mm.ss                         | 13.30.05 |
| IBM USA standard                           | USA          | hh:mm AM or PM                   | 1:30 PM  |
| IBM European standard                      | EUR          | hh.mm.ss                         | 13.30.05 |
| Japanese Industrial Standard Christian Era | JIS          | hh:mm:ss                         | 13:30:05 |
| Site-defined (Local)                       | LOC          | Depends on database country code | —        |

### Notes:

1. In ISO, EUR and JIS format, .ss (or :ss) is optional.
2. In the case of the USA time string format, the minutes specification may be omitted, indicating an implicit specification of 00 minutes. Thus 1 PM is equivalent to 1:00 PM.

3. In the USA time format, the hour must not be greater than 12 and cannot be 0 except for the special case of 00:00 AM. Using the ISO format of the 24-hour clock, the correspondence between the USA format and the 24-hour clock is as follows:
  - 12:01 AM through 12:59 AM corresponds to 00.01.00 through 00.59.00.
  - 01:00 AM through 11:59 AM corresponds to 01.00.00 through 11.59.00.
  - 12:00 PM (noon) through 11:59 PM corresponds to 12.00.00 through 23.59.00.
  - 12:00 AM (midnight) corresponds to 24.00.00, and 00:00 AM (midnight) corresponds to 00.00.00.

#### **Timestamp Strings**

A string representation of a timestamp is a character string that starts with a digit and has a length of at least 16 characters. The complete string representation of a timestamp has the form *yyyy-mm-dd-hh.mm.ss.nnnnnn*. Trailing blanks may be included. Leading zeros may be omitted from the month, day, and hour part of the timestamp, and microseconds may be truncated or entirely omitted. If you choose to omit any digit of the microseconds portion, an implicit specification of 0 is assumed. Thus, 1991-3-2-8.30.00 is equivalent to 1991-03-02-08.30.00.000000.

#### **MBCS Considerations**

Date and timestamp strings must contain only single-byte characters and digits.

---

## **Date and Time Formats**

The character string representation of date and time formats is the default format of datetime values that are associated with the country code of the application. 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 32 on page 352.
- Output Date Format
  - The default output date format is equal to the local date format.

Table 32 shows a listing of the string formats for the various country codes.

*Table 32. 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 |
| 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 |

Table 32. Date and Time Formats by Country Code (continued)

| Country Code                                                            | Local Date Format | Local Time Format | Default Output Date Format | Input Date Formats |
|-------------------------------------------------------------------------|-------------------|-------------------|----------------------------|--------------------|
| 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      |
| 001 USA                                                                 | mm-dd-yyyy        | JIS               | USA                        | LOC, USA, EUR, ISO |
| <b>Note:</b>                                                            |                   |                   |                            |                    |
| (1) Countries using the default C locale are assigned country code 001. |                   |                   |                            |                    |
| (2) yyyy is in Buddhist era: Gregorian + 543 years.                     |                   |                   |                            |                    |





---

## Appendix D. 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:

- Link DataJoiner libraries to certain clients and data sources in AIX
- Use the **db2start** and **db2stop** commands to start and stop Classic Connect processes
- 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

---

### Linking DataJoiner Libraries to Clients and Data Sources in AIX

This section indicates:

- How the method for linking DataJoiner libraries to clients and data sources has changed in Version 2.1.1
- What problem can result when a Version 1.2 application tries to link Version 2.1.1 libraries to certain clients and data sources
- How to resolve this problem

## Change

When you use Version 1.2 in the AIX environment, you edit `djxlink.makefile` and run the `djxlink` shell script to link DataJoiner libraries to Oracle and Sybase client libraries or to DRDA data sources accessed through APPC. When you use Version 2.1.1 in the AIX environment, you do not need to edit `djxlink.makefile` for most data sources. Simply running `djxlink.sh` will link DataJoiner libraries to libraries of nearly all data sources.

## Problem

Running the Version 1.2 `djxlink` shell script in Version 2.1.1 does not work. This script is not shipped with Version 2.1.1.

## Resolution

Run the Version 2.1.1 `djxlink.sh` script. For the small number of data sources that cannot be link-edited using `djxlink.sh`, edit `djxlink.makefile` to contain the data source library information. Then use `djxlink.makefile` to link libraries.

---

## Starting and Stopping Classic Connect Instances

This section indicates:

- How the method for starting and stopping Classic Connect instances has changed in Version 2.1.1
- What problems can result when a Version 1.2 application tries to start and stop Classic Connect instances
- How to prevent this problem

## Change

In Version 1.2, the **`db2start`** command starts Classic Connect processes and the **`db2stop`** command stops these processes. In Version 2.1.1, the **`djxstart`** and **`djxstop`** commands start and stop Classic Connect processes.

## Problem

If, in Version 2.1.1, you issue a **`db2start`** command, the Classic Connect processes will not start. If you issue **`db2stop`**, the Classic Connect processes will not stop.

## Resolution

To start and stop Classic Connect processes for a DataJoiner instance in Version 2.1.1, issue the **`djxstart`** and **`djxstop`** commands.

---

## 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 *SYSEMOTEEUSERS* 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, CPU\_RATIO 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.SYSSSERVERS`. You could also modify an association by updating `SYSIBM.SYSSSERVERS`, and terminate an association by deleting a server name from `SYSIBM.SYSSSERVERS`. 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.SYSSSERVERS`. The changes that you make to the view are propagated to `SYSIBM.SYSSSERVERS`.

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.SYSSSERVERS`, `SYSIBM.SYSREMOTEUSERS`, or any of DataJoiner's other system catalog tables, the statement will fail.

### Resolution

To modify `SYSIBM.SYSSSERVERS` or `SYSIBM.SYSREMOTEUSERS`, use the `SERVER MAPPING` or `USER MAPPING` DDLs, as described in "Change".

---

## Appendix E. Where to Find Out 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 33 lists the DataJoiner, DB2 for CS, and Replication books applicable to installing, configuring, administrating, using, and running applications against DataJoiner. The *DataJoiner for AIX 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 33 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 33).

To understand how the DataJoiner books in Table 33 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 33 document the function and syntax that DataJoiner has *in addition to* the function and syntax that it shares with DB2 for CS.

Table 33 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 books (the *DataJoiner Classic Connect Planning, Installation, and Configuration Guide*, the *DataJoiner Classic Connect data mapper Sample for Windows Installing and Using Guide*, and the *DataJoiner Messages and Problem Determination Guide*) and a program directory.

Table 33. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner

| Book Name                                                                                | Form Number | File Prefix | INF |
|------------------------------------------------------------------------------------------|-------------|-------------|-----|
| <b>DataJoiner Version 2.1.1 Books</b>                                                    |             |             |     |
| <i>DataJoiner for Windows NT Systems Planning, Installation, and Configuration Guide</i> | SC26-9150   | DJXN2       | no  |

This book covers capacity planning, resource management, installation, and configuration tasks for IBM DataJoiner on Microsoft Windows NT operating systems.

---

Table 33. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

| Book Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Form Number | File Prefix | INF |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------|-----|
| <i>DataJoiner for AIX Systems Planning, Installation, and Configuration Guide</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | SC26–9145   | DJXG6       | no  |
| This book covers capacity planning, resource management, installation, and configuration tasks for IBM DataJoiner on AIX 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 Planning, Installation, and Configuration Guide</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | GC26–8869   | DJXC4       | no  |
| This book provides information on 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 in the MVS operating environment: planning your setup; installing components via SMP/E, configuring the kernel, DMSIs, and network communications; managing instances; and creating relational data maps for IMS and VSAM data. |             |             |     |
| <i>DataJoiner Classic Connect data mapper Sample for Windows Installing and Using Guide</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                           | GC26–8873   | DJXZ2       | no  |
| This book provides information on the DataJoiner Classic Connect data mapper sample for Windows. The audience for this information includes system programmers, DBAs, or anyone that needs to produce relational maps (USE grammar) 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 USE grammar statements for input to DataJoiner Classic Connect projection utilities.                                    |             |             |     |



Table 33. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

| Book Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Form Number | File Prefix | INF |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------|-----|
| <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>                                                                                                                                                      |             |             |     |
| DB2 Spatial Extender Administration Guide and Reference                                                                                                                                                                                                                                                                                                                                                                                                                  | 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>                                                                                                                                                                                                                                                                                                                                                                                                                                | SH20-4664   | 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>                                                                                                                                                                                                                         |             |             |     |
| <i>DB2 Administration Guide</i>                                                                                                                                                                                                                                                                                                                                                                                                                                          | S20H-4580   | SQLD0       | yes |
| <p>Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.</p>                                                                                                                                                                                                                                                                                                                      |             |             |     |
| <i>DB2 Database System Monitor Guide and Reference</i>                                                                                                                                                                                                                                                                                                                                                                                                                   | S20H-4871   | SQLF0       | yes |
| <p>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.</p>                                                                                                                                                                                                                                                                                                               |             |             |     |
| <i>DB2 Command Reference</i>                                                                                                                                                                                                                                                                                                                                                                                                                                             | S20H-4645   | SQLN0       | yes |
| <p>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.</p> |             |             |     |
| <i>DB2 API Reference</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                 | S20H-4984   | SQLB0       | yes |

Table 33. DataJoiner, DB2 for CS, and Replication publications applicable to DataJoiner (continued)

| Book Name                                                                                                                                                                                                                                                                                                                                             | Form Number | File Prefix | INF |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------|-----|
| 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 |
| 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 |

Table 33. 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 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. |             |             |     |

---

## 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 33 on page 361.

To view online documentation, follow the instructions located in the README files on the CD-ROM. Most of the books in Table 33 on page 361 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.software.ibm.com/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.software.ibm.com/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 F. 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.software.ibm.com/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.software.ibm.com/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  
500 Columbus Avenue  
Thornwood, NY 10594  
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.

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

W92/H3  
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.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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.

Microsoft, Windows, WindowsNT®, 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, and service names may be trademarks or service marks of others.



---

## Index

### Special Characters

.netrc file and Informix Version 5 131

## A

access

authorizing 229

access, replication

configuring in DataJoiner for AIX 284

accessing data

through DataJoiner DDCCS 10

through Net.Data 9

accessing data sources

configuration overview 69

adding options after installation 43

administration

authorization requirements 274

overview 240

Administrator's Toolkit

command line processor 7

Database Director 7

DB2 Performance Monitor 7

overview 6

Visual Explain 7

ADSTAR Distributed Storage Manager software

requirements 31

after-image columns, description 258

AIX

Apply authentication 293

binding the package 292

installation problems 44

password file 293

setting processes 44

AIX 4.1

deleting options 221

APPC

configuring for DB2RA data sources 107

requirements to use 28

sample values for DB2RA data sources 108

security considerations 113

appc

support for remote clients 191

APPC

template for DB2RA data sources 109

application development

using Net.Data 9

using Software Developer's Kit 8

application requesters, supported 317

Application Server

configuring 207

configuring clients as ARs 213

Apply for AIX

authentication 293

binding the package 292

Apply program

Apply qualifier 248

authorization requirements 275

choosing location 251

control tables, use of 249

instances of 248

introduction to 247

location of 261

password file feature 248

planning considerations 36

processing cycle 251

push vs. pull configuration 261

refresh versus differential refresh 254

relationship to other tables 247

replicating changes 247

run-time processing statements 261

subsetting target columns and rows 259

synchronization with Capture triggers 257

trail tables 250

user ID 275

applying changes to target tables 247

AS/400 access

overview 10

ASN qualifier 305

authentication

end-user for AIX 293

authorization 35

Apply for AIX 293

coordinating 265

for administrator 274

requirements

Apply program 275

Capture program 275

for administration 274

planning 274

system-specific privilege levels 276

authorizing access 229

automating, replication activities 237

## B

before-image columns

column value 258

description 258

NULL value 258

restrictions 266

usage 258

binding

AIX Capture and Apply programs 286

- binding *(continued)*
  - Apply for AIX 292
  - Capture and Apply programs in DB2 292
  - DRDA AS options 308
  - the replication administration tool source, target and control servers 290
- binding packages 97, 129
- books, ordering and viewing 365

## C

- CAE 197
- CALL procedures 261
- Capture and Apply programs
  - binding in DB2 292
  - running 298
- Capture program
  - authorization requirements 275
  - introduction to 243
  - pruning CCD tables 254
  - staging data 254
  - user ID 275
- Capture triggers
  - capturing changes 244
  - control tables, used by 246
  - gaps in target data 265
  - introduction to 244
  - relationships to other tables 244
  - synchronization with the Apply program 257
  - usage 244
- capturing changes, overview 244
- cataloging
  - database directory entry for DB2RA 129
  - database directory entry for DRDA 97
  - DCS directory entry 97
- CCD (consistent change data) tables
  - and the Capture triggers 244
  - description 246, 255
  - internal 257
  - naming schema 305
  - nicknames 306
  - putting at the source 290
  - qualifier 305
  - staging changed data 254
- CD (change data) tables
  - and consistent change data 254
  - pruning of 257
- CD-ROM
  - installing the replication administration tool from 287
- character conversion
  - supported code pages 335
  - when occurs 333
- Client Pack
  - components included 3
  - DB2 Client Application Enabler, overview 3
- Client Pack, components included 3

- clients
  - accessing DataJoiner 25
  - APPC support 191
  - CAE for AIX APPC clients 197
  - CAE for AIX TCP/IP 183
  - CAE for OS/2 TCP/IP 185
  - configuring 175
    - client connections 175
    - DataJoiner and OS/2 clients 199
    - DataJoiner for CAE for AIX APPC 194
    - DataJoiner instance for TCP/IP 177
    - DB2 for CAE for OS/2 V2 199
    - for TCP/IP 178
  - connecting the replication administration tool to DataJoiner 288
  - local 175
  - operating systems supported 4
  - replication
    - Apply for AIX, local 284
    - connecting in AIX 285
    - software developers kit for OS/2 185
    - TCP/IP support 177
- CNTRLSVR.REX 296
- co-existence
  - DB2/6000 and DDCS/6000 219
  - password considerations 220
- code page
  - character conversion 328
  - DataJoiner support 343
  - how determined 328, 338
  - national language support (NLS) 328, 338
  - specifying 343
  - supported values 338
- collating sequence
  - overview of 348
- columns
  - before-image and after-image 258
  - fragmenting 259
  - subsetting 259
- combining data, techniques 260
- commands
  - db2icrt 57
  - db2idrop 228
  - db2iupdt 225
- communication support
  - APPC
    - for DB2RA data sources 107
  - IPX/SPX
    - configuring server 187
    - fileserver 105, 187
    - objectname 105, 188
    - socket number 105, 188
  - server
    - configuring IPX/SPX 187

- communication support *(continued)*
  - SNA
    - for DRDA data sources 78
- Communications Manager/2 (CM/2) 200
- communications support
  - IPX/SPX
    - for DB2RA data sources 104
  - TCP/IP
    - for DB2RA data sources 101
    - for DRDA3 data sources 76
- compilers, supported 30
- concepts, data replication 253
- condensed
  - CCD tables 257
  - staging tables 257
- configuration examples for TCP/IP 179
- configuration parameters
  - fileserv 106, 189
  - ipx\_socket 106, 189
  - IPX/SPX
    - fileserv 105, 187
    - objectname 105, 188
    - socket number 105, 188
  - objectname 106, 189
- configuring
  - APPC for DataJoiner server 114
  - APPC for OS/2 data source 110
  - data sources
    - DB2RA 101
    - DRDA 73
    - IMS and VSAM through CrossAccess 159
    - Informix 131
    - Microsoft SQL Server through Open Client 143
    - Microsoft SQL Server through Visigenic ODBC 149
    - Oracle through Net8 155
    - Oracle through SQL\*Net 155
    - overview 69
    - Sybase through Open Client 143
    - through a generic data access module 165
  - DataJoiner
    - for CAE for AIX APPC clients 194
  - DataJoiner instance for TCP/IP clients 177
- connecting
  - DB2 databases in the replication administration tool 288
  - for replication
    - databases in DataJoiner for AIX 284
  - connecting to the Internet using Net.Data 9
  - Connection affected by ODBC 289
  - connection port 178
  - CONNECTSTRING option 146
  - consistent change data (CCD) tables 244
  - console window 296

- consolidate tables 264
- Control Center
  - managing the replication environment 240
- control server
  - choosing location 251
  - description 231
- control tables
  - Apply trail 250
  - creating 241, 246, 290
  - customizing SQL, description of 242
  - generated by the replication administration tool 241
  - overview of 240
  - pruning 246
  - register tables 246
  - subscription columns 249
  - subscription events 249
  - subscription set 249
  - subscription statements 249
  - subscription targets members 249
  - used by Apply program 249
  - used by Capture triggers 246
- controlling communication support initialization 178
  - DB2COMM 178, 190
  - overview 205
- conversion
  - data types 299
- coordination of
  - target security and source security 265
- country code, DB2CODEPAGE 289
- CREATE DATABASE command
  - national language support (NLS) 327
  - specifying locale information 327
- creating
  - database 61
  - logical unions 260
- creating and running generated SQL
  - order of activities associated with 242
- CrossAccess 159
- customizing
  - SQL 242

## D

- data access modules 70
  - loading dynamically 70
  - overview 69
- data consistency, subscription sets 253
- data replication
  - administrating 239
  - concepts of 253
  - overview 233
  - tools used for 237
- data sources
  - access methods 69
  - CrossAccess 159

- data sources *(continued)*
  - access methods *(continued)*
    - DB2RA 101
    - DRDA 73
    - Informix 131
    - Microsoft SQL Server through Visigenic ODBC 149
    - Oracle Net8 155
    - Oracle SQL\*Net 155
    - Sybase Open Client 143
  - accessed by DataJoiner 27
  - configuration overview 69
- data type conversion
  - DB2 to Informix 300
  - DB2 to Microsoft Jet 302
  - DB2 to MS SQL Server 301
  - DB2 to Oracle 300
  - DB2 to SQL Anywhere 301
  - DB2 to Sybase 301
- database access, two-phase commit 6
- database administration tools, overview 6
- Database Director
  - overview 7
- database manager, not starting 58
- database manager configuration file
  - updating for IPX/SPX 106
- database objects
  - naming rules 347
- databases
  - authorization planning 35
  - creatingDataJoiner 61
- DataJoiner 3, 226
  - as a multivendor server 236
  - classes 367
  - components included
    - Database Director 7
    - DB2 Client Application Enabler 4
    - DB2 Performance Monitor 7
    - Net.Data 9
    - Visual Explain 7
  - configuring for TCP/IP clients 182
  - data type conversions 299
  - differences between versions 355
  - for AIX
    - connecting clients for replication 285
    - connecting databases for replication 284
    - creating databases for replication 284
    - installing replication 283
  - nicknames, use of 243
  - overview 3
  - overview of 243
  - platforms supported 4
  - programs
    - DB2 Client Application Enabler 3
- DataJoiner *(continued)*
  - programs *(continued)*
    - Net.Data 3
    - Software Developer's Kit 3
  - starting at system boot 226
  - the replication administration tool as a client 288
  - user mappings 277
- DataJoiner DDCS
  - overview 10
- DataJoiner WWW site 365
- DataPropagator NonRelational 233
- DataPropagator Relational V1, support for DataJoiner 233
- date
  - definition of 348
  - formats 351
  - strings 349
- datetime values
  - overview of 348
  - string representations 349
- DB2
  - considerations 219
  - environment variables 58
  - log 244
- DB2/6000 considerations 219
- DB2 Client Application Enabler
  - accessing databases 4, 5
  - platforms supported 4
- DB2 log 244
- DB2 Performance Monitor
  - overview 7
- DB2 replication
  - gaps in target data 265
- DB2 WWW site 365
- DB2CODEPAGE setting from the replication
  - administration tool 289
- DB2COMM 205
- db2cshcr script 57
- DB2GROUPS setting 229
- db2icrt command 57
- db2idrop 228
- db2iupdt 225
- db2profile script 57
- DB2RA data sources
  - accessing 101
  - using nicknames and pass-through 101
    - cataloging a node directory 125
    - configuring APPC 107
    - configuring IPX/SPX for 104
    - configuring TCP/IP for 101
    - testing connections 128
    - updating catalog views 125
- db2schr
  - running 60



- DBA Utility font 57
- DBCS, not supported for replication 267
- DBLIB connections
  - improving performance for 251
- DDCS
  - considerations 219
- DDCS, using for connectivity 239
- DDCS/6000 considerations 219
- deleting options
  - AIX 3.2 220
  - AIX 4.1 221
- developing applications
  - using Net.Data 9
  - using Software Developer's Kit 8
- differential refresh 231, 254
- disk requirements 22
- distributed database connection services (DDCS) 239
- distributed-unit-of-work access 6
- DJX\_NR\_CONFIG environment variable 59
- DJXCOMM environment variable 58, 70
- djxlink.makefile
  - description 47
  - warning messages during link 53
- djxlink.sh 47
- double-byte character set user
  - code page support 341
- double-byte character string, not supported for replication 267
- DPROP NR (DataPropagator NonRelational) 233
- DRDA
  - application requesters, supported 317
  - Application Server
    - bind options 308
    - CCSID support 312
    - configuring 312
    - DB2 for VM, considerations 318
    - description 307
    - PTFs for DB2 for MVS 317
    - PTFs for DB2 for VM 317
    - PTFs for DDCS 317
    - PTFs for OS/400 318
    - supported ARs 317
    - supported functions 307
  - databases software requirements 26
  - SNA system defaults 90
- DRDA data sources
  - accessing 73
  - Application Server 207
  - outside the heterogeneous environment 128
    - binding DataJoiner packages 129
    - cataloging a database directory entry 129
    - configuring communications for 128
    - updating the node directory 129
  - using DataJoiner DDCS 95

- DRDA data sources (*continued*)
  - binding DataJoiner packages 97
  - cataloging a database directory entry 97
  - cataloging a DCS directory entry 97
  - configuring communications for 96
  - updating the node directory 96
- using nicknames and pass-through 73
  - configuring SNA Server/6000 78
  - configuring TCP/IP for 76
  - testing connections 95
  - updating catalog views 93
  - updating the node directory 92

## E

- editing the replication administration tool
  - logic 296
  - SQL 297
- electronic information 365
- encryption 35
- environment variables 57
  - DB2COMM 205
  - DB2COMM and APPC 114
  - description and syntax 58
  - DJX\_NR\_CONFIG 59
  - DJXCOMM 58
  - LIBPATH 219
  - setting 57
  - used by DataJoiner 223
- errors
  - SQL1224N 60
- errors, possible
  - warnings during djxlink.makefile 53
- ESQL/C library 138

## F

- filesaver
  - configuration parameter 106, 189
  - definition 105, 187
- fixed-disk space required 21
- fixes, code 220
- fragmentation
  - horizontal 259
  - vertical 259
- fragmenting target tables 259
- full refresh
  - description 231
- full refresh description 254

## G

- gaps in target data 265
- general library information 361
- generated SQL
  - editing 297
  - running 298

- generic data sources
  - accessing through a generic data access module 165
- graphical user interface
  - software requirements 29
- graphical user interface (GUI)
  - replication administration tool's
    - creating control tables from 241
    - overview 239
    - using for administration 239

## H

- HACMP, software requirements 30
- hardware requirements 21
- heterogeneous server 231
- horizontal subsetting
  - description 259
- hosts file 179

## I

- IBM replication solution
  - restrictions 266
- IBM Replication Solution
  - tools 233
- IMS
  - accessing through CrossAccess 159
  - required client software 28
- Informix
  - accessing 131
  - esql/rt libraries 137
  - required client software 28
  - setting DJXCOMM for 140
  - Version 5 data access module 131
  - Version 7 data access module 137
- installation
  - authorization 35
  - common AIX problem 44
  - password encryption 35
  - prerequisite software 35
  - procedure 35
  - PTF 220
  - replication
    - Apply for AIX 283
    - DataJoiner for AIX 283
  - steps
    - for DataJoiner 44
  - the replication administration tool 287
- installation steps
  - creating a DataJoiner database 61
  - creating an instance 55
  - linking DataJoiner to data sources 47
  - restoring product files 45
  - setting AIX processes 60
  - setting environment variables 57
  - setting up password encryption 47

- instances
  - authorizing access within the database manager 229
  - configuring for TCP/IP clients 177
  - creating 55
  - memory requirements 21
  - multiple 226
  - name 57
  - owner 55
  - recycling 55
  - removing 228
  - setup for replication
    - DataJoiner for AIX 284
  - starting 61
  - starting and stopping 223
  - starting at system boot 226
  - stopping and restarting 223
- INSTHOME 57
- internal CCD tables
  - reducing network load 257
- Internet information 365
- interrupt port 178
- introduction to the replication administration tool
  - Apply program 247
  - Capture program 243
  - Capture trigger 244
- ipcrm command, stopping instances 223
- ipcs command, stopping instances 223
- ipx\_socket
  - configuration parameter 189
- ipx\_socket configuration parameter 106
- IPX/SPX
  - configuring for DB2RA data sources 104
  - configuring server 187
  - fileserv 105, 187
  - objectname 105, 188
  - setting DB2COMM 107
  - socket number 105, 188
  - updating the database manager configuration file 106

## J

- joins
  - description 265
  - multitable 260
  - of tables 264

## K

- kill command, stopping instances 223

## L

- LAN DLC Adapter Profile 201
- levels, privileges 276
- LIBPATH environment variable 219
- library information, general 361

- library path 219
- linking to data source client software
  - description 47
  - using data access modules 70
  - using DRDA Application Requester 73
  - warning messages during 53
- local client
  - Apply for AIX 284
  - description 175
- logic
  - CNTRLSVR.REX 296
  - editing the replication administration tool 296
  - SRCESVR.REX 296
  - TARGSVR.REX 296
  - TBLSPACE.REX 296
- logical partitioning key and replication 260
- logical unions, creating 260
- LU 6.2 88
- LU sessions 294

## M

- maintaining DataJoiner 220
- managing databases using Database Director 7
- mapping user IDs 277
- maxuproc parameter 60
- memory requirements 21
- Microsoft SQL Server 145
  - accessing through Open Client 143
  - accessing through Visigenic ODBC 149
  - improving performance for 251
  - required client software 28
  - support 50
- monitoring databases
  - using DB2 Performance Monitor 7
- multiple instances 43, 226
- multisite union 265
- multitable joins and unions 260
- multivendor access 236

## N

- naming rules
  - user names and group names 56
- national language support (NLS)
  - character sets 345
  - client support 333
  - code page 338
  - code pages 333
  - code sets supported 338
  - codesets supported 333
  - country code 338
  - country codes 333
  - DataJoiner code page support 343
  - datetime values 348
  - host CCSID 333
  - locale 333

- national language support (NLS) *(continued)*
  - overview of 327
  - territories supported 333
- Net.Data
  - overview 9
- Net8 155
- network options 179
- node directory
  - updating for DB2RA data sources
    - heterogeneous access 125
    - outside the heterogeneous environment 129
  - updating for DRDA data sources
    - heterogeneous access 92
    - using DataJoiner DDCCS 96
- non-IBM database
  - CCD tables and Capture triggers 305
  - nicknames in DataJoiner 306
- non-IBM server
  - description 231
- non-IBM database
  - control tables 305
  - register control table 306
  - table names and qualifiers 305
- notices 371

## O

- object file 47
- objectname
  - configuration parameter 106, 189
  - definition 105, 188
- ODBC affecting Preferences 289
- Open Client 143
- operating system requirements 25
- options
  - adding 220
  - AIX network 179
  - deleting 220
  - on AIX 3.2 220
  - on AIX 4.1 220
- Oracle
  - accessing through Net8 155
  - accessing through SQL\*Net 155
  - required client software 28
  - section of makefile 50
  - tnsnames.ora file 155
- ordering publications 365
- OS/2
  - configuring DataJoiner and OS/2 clients 199
  - configuring DB2 for CAE for OS/2 V2 199

## P

- packages
  - binding 97, 129
- partitioning key and replication 260

- password file
  - Apply for AIX 293
  - authorization requirements 276
  - using 248
- passwords
  - encryption planning 35
  - OS/2 client 220
  - setting up additional encryption 47
- passwords, coordination of 265
- performance, improving 251
- planning
  - authorization requirements 274
  - replication considerations 36
- point-in-time target tables 253
- port
  - connection 178
  - interrupt 178
  - numbers 178
  - planning 178
- Preferences affected by ODBC 289
- Preferences notebook, setting the replication
  - administration tool 289
- prerequisite software 35
- primary key
  - errors when updating 259
  - point-in-time target table 253
  - user copy target table 253
- printing publications 365
- privilege levels 276
- products supported
  - as sources 237
  - as targets 237
- profiles
  - control point 82, 117
  - for OS/2 V2 clients 200, 203
  - link station 116
  - LU 6.2 local LU 84, 119
  - LU 6.2 side information 89, 123
  - mode 88, 121
  - partner LU 6.2 location 86, 120
  - token-ring link station 81
  - token-ring SNA DLC 80, 115
  - TPN 195, 208
  - TPN for AIX client 196, 210
- Program Temporary Fixes (PTFs) 220
- promote function 265
- PRTCT 294
- pruning
  - CCD tables 254
  - CD and UOW tables 257
  - control tables 246
- publications 365
- push and pull Apply program configuration
  - choosing a configuration 264

- push and pull configuration
  - comparison of 261
  - description of 261

## R

- recycling instances 223
- refresh, the Apply program 243, 247
- refresh copying 254
- register control table 306
- register tables 246
- remote TCP/IP clients 177
- removing instances 228
- replica 253
- replica target tables 264
- replication
  - Apply program 247
  - Capture and Apply program requirements 273
  - Capture program, introduction to 243
  - Capture triggers, introduction to 244
  - connecting all databases 287
  - description 231
  - differential refresh, description 231
  - full refresh, description 231
  - hardware requirements 270
  - objects, description of 241
  - overview 233, 239
  - overview of components 13
  - performance tip for 257
  - planning considerations 36
  - replication administration tool 237
  - set up in the replication administration tool 287
  - setting up
    - DataJoiner for AIX 283
    - software requirements 270
  - sources 241
  - subscription sets 241
  - supported environments
    - DataJoiner 234
    - DB2 family 234
    - IMS and VSAM 234
    - Lotus NotesPump 234
    - Non-DB2 234
    - non-DB2 236
  - supported sources and targets 269
  - tables 241
  - transaction based 257
  - update anywhere 232, 264
- replication administration tool
  - Apply program, and the 247
  - creating control tables 240
  - customizing replication requests 242
  - GUI
    - creating control tables from 241
  - introduction to 237

- replication administration tool (*continued*)
  - planning considerations 36
- replication targets, sets 232
- restarting instances 223
- restoring product files to target machine 45
- restrictions
  - in using the IBM replication solution 266
  - with before-image columns 266
- rows, subsetting 259
- run-time processing, SQL statements 261
- running Capture and Apply programs 298
- running the replication administration tool sql 298

## S

- SECACPT 294
- security 35, 274
  - APPC considerations 113
  - description 265
  - general 35
- sequence for the replication administration tool SQL 295
- servers
  - binding from the replication administration tool 290
  - control 231
  - non-IBM 231
  - source 232
  - target 232
- service\_name 178
- SET SERVER OPTIONS statement
  - CONNECTSTRING option 146
- setting up
  - replication
    - DataJoiner for AIX 283
- setup
  - finalizing Apply for AIX 292
- SNA
  - configuring for DRDA data sources 78
  - local profile 80, 119
  - LU 6.2 side information 88, 122
  - section of djxlink.makefile 50
  - system defaults 124
    - for APPC 124
    - for DRDA 124
  - system defaults for DRDA 90
- SO\_KEEPALIVE 179
- socket number 188
- socket number definition 105
- Software Developer's Kit
  - overview 8
- software developers kit for AIX 183
- software requirements 25
  - ADSTAR Distributed Storage Manager 31
  - DataJoiner data sources 27
  - DRDA databases 26
  - graphical user interface 29

- software requirements (*continued*)
  - HACMP 30
  - operating system 25
  - software developers kits (SDKs) 25
  - supported compilers 30
  - transaction managers 29
- solution, data replication 233
- source
  - control tables, on the 246
  - description 232, 241
  - security, coordination with target 265
  - supported 237
  - table
    - definition of 239
    - views 264
- source server
  - description 232
- sources and targets
  - supported for replication 269
- SQL
  - before and after, description 261
  - creating and running of generated 242
  - editing the replication administration tool 297
  - generated
    - deferring the running of 242
    - description 242
    - order of creating and running 242
    - running the replication administration tool 298
  - statements, run-time processing of 261
- SQL access plans, tuning 7
- SQL access plans, viewing using Visual Explain 7
- SQL\*Net 155
- SQL Server (Microsoft) 145
  - support 50
- SQL1224N error 60
- SRCESVR.REX 296
  - CCD table name 305
- staging data description 254
- staging table
  - complete 257
  - description of 254
- starting and stopping instances 223
- starting Capture and Apply programs 298
- starting the replication administration tool 295
- stopping and restarting instances 223
- storage requirements 21
- subscription
  - columns tables description 249
  - events tables description 249
  - member
    - description of 239
    - member, description of 253
    - members description 232

- subscription (*continued*)
  - set
    - and the Apply qualifier 248
    - description 241
    - description of 239
    - tables, description of 249
  - sets
    - description of 253
    - run-time processing statements 261
    - target views 265
  - statements tables description 249
  - targets members tables description 249
- subscription set
  - description 232
- subsetting
  - horizontal 259
  - source tables 259
  - vertical 259
- supported sources and targets
  - for replication 269
- Sybase
  - accessing through Open Client 143
  - improving performance for 251
  - required client software 28
  - section of djxlink.makefile 50
  - specifying a connect string 146
- synchpoint 257
- synchronization of
  - target security and source security 265
- system configuration
  - using DataJoiner 4, 5
  - using DataJoiner DDCS Multi-User Gateway 11
  - using Net.Data 9

**T**

- table space, specifying in logic 297
- target
  - description 241
  - security, coordination with source 265
  - supported 237
- tables
  - CCD description 255
  - fragmenting of 259
  - point-in-time 253
  - pruning of 257
  - replica 253
  - subsetting columns in 259
  - subsetting rows in 259
  - types of 253
  - user copy 253
  - views 265
- target server 232
- TARGSVR.REX 296
- TBLSPACE.REX 296

- TCP/IP
  - configuration examples 179
  - configuring a DataJoiner instance 182
  - configuring for DB2RA data sources 101
  - configuring for DRDA3 data sources 76
  - requirements to use 28
  - support for remote clients 177
  - template for DB2RA data sources 104
  - template for DRDA3 data sources 77
  - warning for client-server support 179
- tcp\_keepidl 179
- tcp\_keepintvl 179
- templates
  - AIX data source planning 76
  - APPC client planning 194
  - TCP/IP planning 182
- territory, relationship to locale 312
- the replication administration tool
  - authorization requirements 274
  - console window 296
  - creating control tables 290
  - hardware requirements 270
  - initial startup 295
  - installing 287
  - setting preferences 289
  - software requirements 270
  - using 295
- time
  - definition of 349
  - formats 351
- time strings
  - definition of 350
- timestamp
  - definition of 349
- timestamp strings
  - definition of 351
- token-ring
  - CM/2 profile for CAE for OS/2 201
  - configuring SNA Server/6000 80
  - with SNA Server for APPC 115
- tools
  - used for data replication 237
- trademarks 373
- transaction manager software requirements 29
- transaction program (TP) definition 112
- triggers
  - See also* Capture triggers 290
- two-phase commit
  - configuring 169
- two-phase commit database access 6

**U**

- unions
  - multisite 265
  - multitable 260

- unit-of-work (UOW) tables
  - and the Capture triggers 244
  - pruning of 257
- update-anywhere replication
  - description 232, 264
- updating DataJoiner 220
- upgrade 43
- URL
  - <http://www.software.ibm.com/data/datajoiner/> 50
- user copy target tables 253
- user IDs
  - mapping 277
  - specified in the replication administration tool 289
- using the replication administration tool 242, 295

## V

- variables, setting environment 57
- VERIFY 294
- vertical subsetting 259
- viewing publications 365
- views
  - definitions 264
  - registering as a source table 260
  - source, description 264
  - target, description 265
  - used as sources 264
  - using to perform joins 264
- Visigenic
  - ODBC driver 149
  - odbc.ini file 150
- Visual Explain
  - overview 7
- VSAM
  - accessing through CrossAccess 159
  - required client software 28

## W

- warning messages during link 53
- wildcards 296
- Windows NT
  - connecting to DB2 databases for replication 288
- WWW information 365

## X

- XA libraries
  - link-editing for two-phase commit 169





---

# Readers' Comments — We'd Like to Hear from You

**DB2 DataJoiner® for AIX Systems  
Planning, Installation,  
and Configuration Guide  
Version 2 Release 1 Modification 1**

**Publication No. SC26-9145-01**

**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.



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



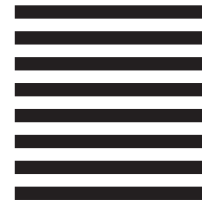
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# 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

Cut or Fold  
Along Line





Part Number: CT7PMIE



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

SC26-9145-01



CT7PMIE



Spine information:



DB2 DataJoiner<sup>®</sup> for AIX Systems **Planning, Installation, and  
Configuration Guide**

Version 2  
Release 1  
Modification 1

SC26-  
9145-01