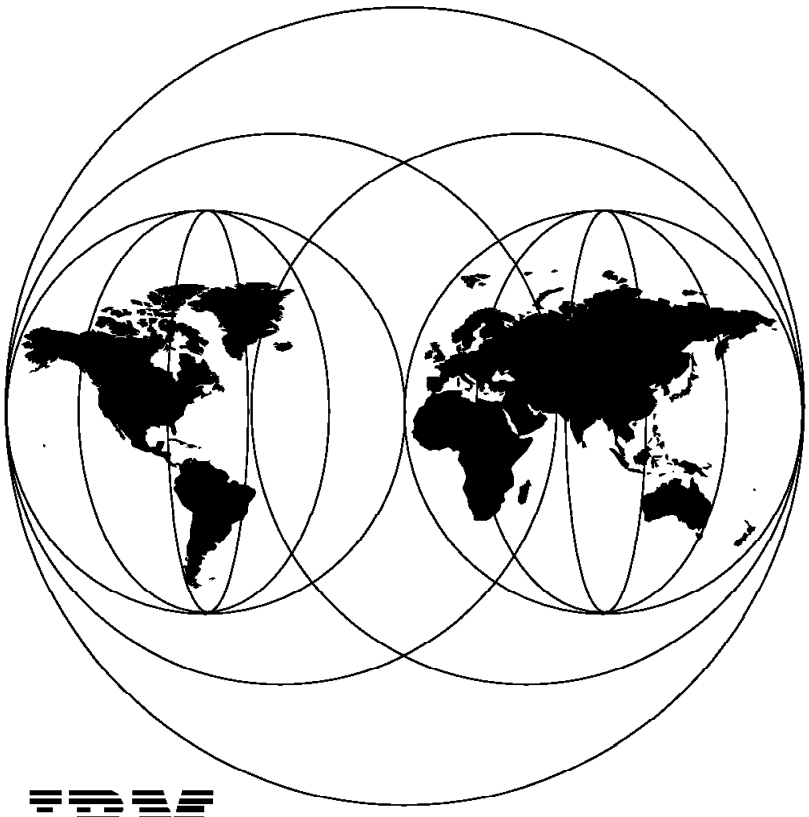
International Technical Support Organization

SG24-2566-00

DataJoiner Implementation and Usage Guide

October 1995



IBM

International Technical Support Organization
San Jose Center

International Technical Support Organization

SG24-2566-00

DataJoiner Implementation and Usage Guide

October 1995

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xv.

First Edition (October 1995)

This edition applies to Version 1 Release 1 of IBM DataJoiner for AIX, 5696-DJX.

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

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 471 Building 070B
5600 Cottle Road
San Jose, California 95193-0001

When you send information to IBM, you grant IBM a non-exclusive 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. 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.

Abstract

This document provides details of how to install the IBM DataJoiner product and related supporting software in a variety of environments, and use DataJoiner to accomplish many specific tasks. The purpose is to provide information on *how* these tasks are to be accomplished. This book includes and describes specific configurations and recommendations. Many of these recommendations were actually implemented with DataJoiner on the supported operating platforms.

This document was written for database and system administrators responsible for planning and implementing data access strategies to satisfy a variety of business requirements. General understanding of relational databases is assumed. Some knowledge of AIX/6000, DB2 for AIX/6000, and related communication software is assumed for readers who will plan for and install DataJoiner. Some knowledge of the target database system and its operating environment is assumed for readers who will use DataJoiner to access data from remote sources.

(277 pages)

© Copyright IBM Corp. 1995

Contents

© Copyright IBM Corp. 1995

Figures

© Copyright IBM Corp. 1995

Tables

© Copyright IBM Corp. 1995

Special Notices

This publication is intended to help database administrators and system administrators who need to manage and provide access to multiple database management systems (DBMSs). The IBM product, DataJoiner for AIX, provides transparent access to multiple DBMSs using global optimization and other very useful capabilities. The information and examples in this publication can help the data and system administrators to both establish the environment and use the functions of DataJoiner. This book can also be useful to anyone who is evaluating DataJoiner or learning about its capabilities. The information in this publication is not intended as the specification of any programming interfaces that are provided by DataJoiner. See the PUBLICATIONS section of the IBM Programming Announcement for DataJoiner for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program, or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

Advanced Peer-to-Peer Networking	AIX
AIX/6000	APPN
AS/400	DATABASE 2
DATABASE 2 OS/400	DB2

© Copyright IBM Corp. 1995

DB2/2	DB2/400
DB2/6000	Distributed Database Connection Services/2
Distributed Relational Database Architecture	DRDA
IBM	IMS
IMS/ESA	MVS/ESA
Operating System/2	Operating System/400
OS/2	OS/400
PS/2	RACF
RISC System/6000	S/390
SQL/DS	SQL/400
System/390	Virtual Machine/Enterprise Systems Architecture
VM/ESA	VSE/ESA
VTAM	

The following terms are trademarks of other companies:

HP/UX	Hewlett-Packard Company
Netware, Novell	Novell, Incorporated
Oracle, Oracle 7	Oracle Corporation
SunOS, SPARCstation, Network File System, NFS	Sun Microsystems, Inc.
Sybase	Sybase, Incorporated

Windows is a trademark of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This document shows how DataJoiner can be used to meet many requirements for access to and management of data among and across several different database platforms, both relational and nonrelational. It contains specific recommendations and examples for establishing connectivity to the different platforms, defining data sources and targets to the DataJoiner system, establishing access to the different sources and targets, and using DataJoiner and the data available through DataJoiner with several client environments. Most of the information is in the form of specific scenarios to show a way of implementing a particular solution. The scenarios are based on actual implementations carried out to support the development of this book.

This document is intended for database administrators, system administrators, and other information technology professionals responsible for developing and deploying solutions that involve access to multiple data management or database management systems. This book can also be useful if you are evaluating DataJoiner for its usefulness and value in your environment. Chapter 1, “Introduction and Overview” on page 1 and some of the publications listed in “Related Publications” on page xix can be useful in learning about the capabilities of IBM products that support a heterogeneous data management environment.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction and Overview” provides a brief introduction to the functions and capabilities of the DataJoiner product. The chapter also describes at a high level the environment we used to implement the systems employed in developing the examples for this book.
- Chapter 2, “Planning and Installation” provides information on how to set up the environment for DataJoiner and considerations for prerequisite products. The chapter then describes how to install DataJoiner, using examples that were actually implemented.
- Chapter 3, “Configuration of DataJoiner for Clients” provides information and examples on what you need to define to DataJoiner to enable clients of various types to access it and the data that it is managing.
- Chapter 4, “Configuration of DataJoiner Clients” describes methods for and shows examples of setting up the client systems to access DataJoiner.
- Chapter 5, “Configuration of DataJoiner for Data Sources” provides information on defining the various databases that DataJoiner is to manage, to the system on which DataJoiner is installed and to DataJoiner itself.
- Chapter 6, “Security” shows how to administer security in the DataJoiner environment. The chapter includes six specific scenarios that illustrate how to implement security with different combinations of clients and target databases.
- Chapter 7, “Using DataJoiner” provides information on how users actually access the data managed by DataJoiner. The chapter provides information

© Copyright IBM Corp. 1995

on the steps to enable users to get to the data, and various structured query language (SQL) capabilities that can then be used.

- Chapter 8, " DataJoiner Capacity and Performance" provides some tips, techniques, and examples for using various tools to understand and improve performance.
- Several appendixes are provided for reference, and include such information as configuration listings for the various systems used in the implementation that supported the development of this book. The appendixes are:
  - Appendix A, "AIX Definitions in Germany"
  - Appendix B, "VSE/ESA Definitions in Germany"
  - Appendix C, "VM/ESA Definitions in Germany"
  - Appendix D, "MVS VTAM Definitions in San Jose"
  - Appendix E, "AIX Definitions in San Jose"
  - Appendix F, "CAE/2 Configuration File for APPC Connection"
  - Appendix G, "DB2/2 v.1.x Configuration File for APPC Connection"
  - Appendix H, "DB2/2 v.2.1 Configuration File for APPC Connection"
  - Appendix I, "CAE/6000 Configuration File for APPC Connection"
  - Appendix J, "TCP/IP Correlations and Worksheets"
  - Appendix K, "APPC (SNA) Correlations and Worksheets"
  - Appendix L, "AIX Definitions in the Severn System"

Product Nomenclature and Abbreviations

The following table lists the full names of some of the software products referred to in this publication and, for each, lists the commonly used short names or abbreviations. This publication often uses the short names.

Full Product Name	Short Name or Abbreviation
IBM DataJoiner for AIX	DataJoiner
IBM DATABASE 2	DB2 The term *DB2* by itself is often used to refer to the DB2 family of IBM relational database products, and sometimes to the DB2 for MVS product.
DB2 for MVS	DB2/MVS or DB2
DB2 for OS/2	DB2/2
DB2 for AIX/6000	DB2/6000
DB2 for OS/400	DB2/400
DB2 for VM and VSE	DB2/VM if referring to an installation in the VM environment, DB2/VSE if referring to an installation in the VSE environment. (DB2 for VM and VSE was formerly known as SQL/DS.)
DB2 for HP-UX	DB2/HP or DB2/HP-UX

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *AIX SNA Server: User's Guide*, SC31-7002
- *DataJoiner Planning, Installation, and Configuration Guide*, SC26-8244
- *DB2/6000 Command Reference*, SC09-1575
- *IBM DataJoiner Administration*, SC26-8245
- *IBM DataJoiner Application Programming and SQL Reference Supplement*, SC26-8330
- *IBM DataJoiner Generic Access API Reference*, SC26-8246
- *IBM DataJoiner An Introduction to DataJoiner*, GC26-8243
- *IBM DataJoiner Messages and Codes*, SC26-8331
- *SQL Formal Register of Extensions and Differences*, SC26-3316

International Technical Support Organization Publications

- *DB2/6000 Client/Server Usage Guide*, GG24-4322
- *Distributed Relational Database Cross Platform Connectivity and Application*, GG24-4311
- *DRDA Client/Server Application Scenarios for VM and VSE*, GG24-4193
- *The IBM Xstation Handbook*, GG24-3695

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in: *International Technical Support Organization Bibliography of Redbooks*, GG24-3070.

To get a catalog of ITSO technical publications (known as "redbooks"), VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

International Technical Support Organization on the World Wide Web

Internet users can find additional material about new redbooks on the ITSO World Wide Web home page. Point your web browser to the following Uniform Resource Locator (URL):

http://www.redbooks.ibm.com/redbooks

IBM internal users can also download redbooks or scan through redbook abstracts. Point your web browser to the internal IBM Redbooks home page:

http://w3.itsc.pok.ibm.com/redbooks/redbooks.html

If you do not yet have World Wide Web access, you still can obtain the list of all current redbooks, since this document is also available through Internet by anonymous FTP to

```
ftp.almaden.ibm.com
cd /redbooks
get itsopub.txt
```

All users of ITSO publications are encouraged to provide feedback to improve quality over time. A feedback form is in the back of this redbook. Questions and feedback to redbooks can also be sent to

REDBOOK at WTSCPOK *or*
REDBOOK@VNET.IBM.COM *or*
USIB5FWN at IBMMAIL

Acknowledgments

This project was designed and managed by:

Don Cameron
IBM International Technical Support Organization, San Jose Center

Manfred Mittelmeier
IBM International Technical Support Organization, Böblingen, Germany Center

The authors of this document are:

Marcio José Barbosa
IBM Brasil

Bernhard Bühler
IBM Germany

Don Cameron
IBM International Technical Support Organization, San Jose Center

Deirdre Clyne
IBM Ireland

Brian Davidson
IBM Canada

Erick Garcia Reza
IBM Mexico

Wilhelm Mild
IBM Germany

Manfred Mittelmeier
IBM International Technical Support Organization, Böblingen, Germany Center

This publication is the result of a residency conducted at the International Technical Support Organization, San Jose Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Joe DeCarlo
International Technical Support Organization, San Jose Center

Guido De Simoni
International Technical Support Organization, San Jose Center

Susan Gausden
Independent Consultant
Montreal, QC, Canada

Scott Howard
IBM Education and Training

Josette Huang
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Jim Kleewein
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Terry Mason
Independent Consultant
Montreal, QC, Canada

Bob Picciano
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Silvio Podcameni
International Technical Support Organization, San Jose Center

Dave Tolleson
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Yun Wang
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Andy Yang
IBM Santa Teresa Laboratory, San Jose, California U.S.A.

Thanks also to Shirley Hentzell for editing this book, and to Stephanie Manning for her editorial assistance.

Chapter 1. Introduction and Overview

This chapter introduces IBM DataJoiner Version 1 and introduces the environment used in the development of this book. The introduction to DataJoiner briefly describes its capabilities, multiserver support, client support, and engine functions.

1.1 DataJoiner Version 1 Introduction

DataJoiner is a multidatabase server that provides access to a wide range of data sources: IBM DB2 Family, other relational databases such as Sybase and Oracle, as well as nonrelational database systems. From the point of view of an application developer, all data resources are being provided from a single database management system (DBMS). A standards-compliant structured query language (SQL) interface is provided from OS/2, AIX, DOS, Windows, HP-UX and Solaris client systems to simplify the application development process. Applications are further simplified since the client applications need only connect to the DataJoiner server. DataJoiner handles all session connections and security considerations to reach the remote databases.

DataJoiner provides network and location transparency for the user. All data can be accessed as if the source were local. Users do not need to know the physical location of the data. Furthermore, SQL dialects, data access methods, networking protocols, operating systems, data types, error codes, and functional differences also become transparent to the application.

DataJoiner also enables new decision-support applications by providing a heterogeneous join across dissimilar data sources with a single data-access request. DataJoiner is unique among decision-support enabling software in that an SQL statement, either embedded SQL or SQL call level interface (CLI), can be *optimized* by the DataJoiner server. That is, the optimal or ideal access path to the remote data is determined by the cost-based optimizer engine. Therefore the application developer is relieved of the responsibility of understanding and coding specifically to a remote database manager. For example, DataJoiner may select the sequence of data access or the location at which to perform a ″join″ based upon

- The amount of data that will qualify from a specific database
- The join approaches available to use at a specific database
- The relative performance of a particular hardware platform
- The speed of the communications between the platforms.

DataJoiner enables database vendor independence by not requiring that applications write to the specific application programming interface (API) of every remote DBMS. The developer of the client application can code the data access requests as if the data existed in a single DBMS, that is within a DataJoiner DBMS. Not only does DataJoiner make database vendor independence possible, it can actually assist in data migration. Data can be transferred, a table at a time, at the organization′s own pace, with minimal impact on existing applications. This migration method is best used on small to medium-sized tables.

DataJoiner can access data stored in any IBM DB2 relational database (including DB2 for HP-UX and DB2 for the Solaris Operating Environment), several other

© Copyright IBM Corp. 1995

popular relational DBMSs like Oracle and Sybase, as well as several nonrelational data sources such as Virtual Storage Access Method (VSAM) and Information Management System (IMS) DB.

The SQL dialect provided supports all of the data manipulation language (DML) statements. Any use of data definition language (DDL) and data control language (DCL) is expected to be performed through other data management tools and utilities. DDL and DCL can also be invoked using a passthrough mode that will permit use of functions not supported directly by DataJoiner.

1.1.1 DataJoiner Multiserver Support

DataJoiner provides access, via the Distributed Relational Database Architecture (DRDA) application requester function, to IBM DB2 family databases on MVS, VM, or OS/400 platforms. In addition, DataJoiner supports access to DB2 family databases running on OS/2, AIX, HP-UX, and Sun Solaris platforms using the *JRA protocol*. The JRA protocol refers to the DB2 format and protocol between clients and servers.

DataJoiner can also access data on a remote Oracle relational database through SQL*NET. Access to the Sybase relational database is through Sybase Open Client. As far as Sybase and Oracle are concerned, DataJoiner appears to be a regular AIX-based client application. Therefore DataJoiner can access all platform/protocol combinations that any Sybase or Oracle AIX application can access.

Access to nonrelational data on IBM platforms is currently provided by third-party gateways.

DataJoiner also provides a call-level driver interface that permits the creation of data access modules for any DBMS not directly supported by DataJoiner. This call-level interface is similar in function to Microsoft's Open Database Connectivity (ODBC) call-level interface. Any X/Open or ODBC driver that supports CORE level functions will work with DataJoiner.

1.1.2 DataJoiner Client Support

DataJoiner supports the same client platforms and protocols as those supported by DB2 for AIX (DB2/6000) Version 1. The platforms are DOS, DOS/Windows, AIX, OS/2, HP-UX, and Solaris. Communication protocols are Transmission Control Protocol/Internet Protocol (TCP/IP), Advanced Program-to-Program Communication (APPC), and IPX/SPX which requires NOV*IX (from FireFox Inc.) for NetWare. Not every platform supports every protocol. Client access is made possible by way of IBM's Client Application Enabler (CAE) products, which support embedded SQL and CLI applications. Windows applications can also access DataJoiner through the ODBC driver which is part of the CAE DOS product.

DataJoiner also supports some clients running other than the latest versions. Examples include Extended Services/2 and the combination of Communications Manager/2 and DB2 for OS/2 Version 1.0. See the *DataJoiner Planning, Installation, and Configuration Guide* for details on client software supported.

DataJoiner provides a consistent SQL dialect to all client applications and tools. This dialect is identical to that provided by IBM DB2 for AIX (DB2/6000) Version 1. The full set of DataJoiner DML may be used with all of the target data

sources. DataJoiner translates the DML to the native dialect of a supported remote data source when necessary.

A passthrough mode is provided to enable the exploitation of specific remote database functions that are not native to DataJoiner. For example, if an end user or application programmer needs to execute the Oracle standard deviation function, this can be accomplished by issuing the set passthru command to Oracle andd issuing the Oracle standard deviation function. However, in this mode, DataJoiner does not optimize the SQL request but rather sends it directly to the remote DBMS.

Figure 1 shows the breadth of DataJoiner's support for clients and remote data sources.

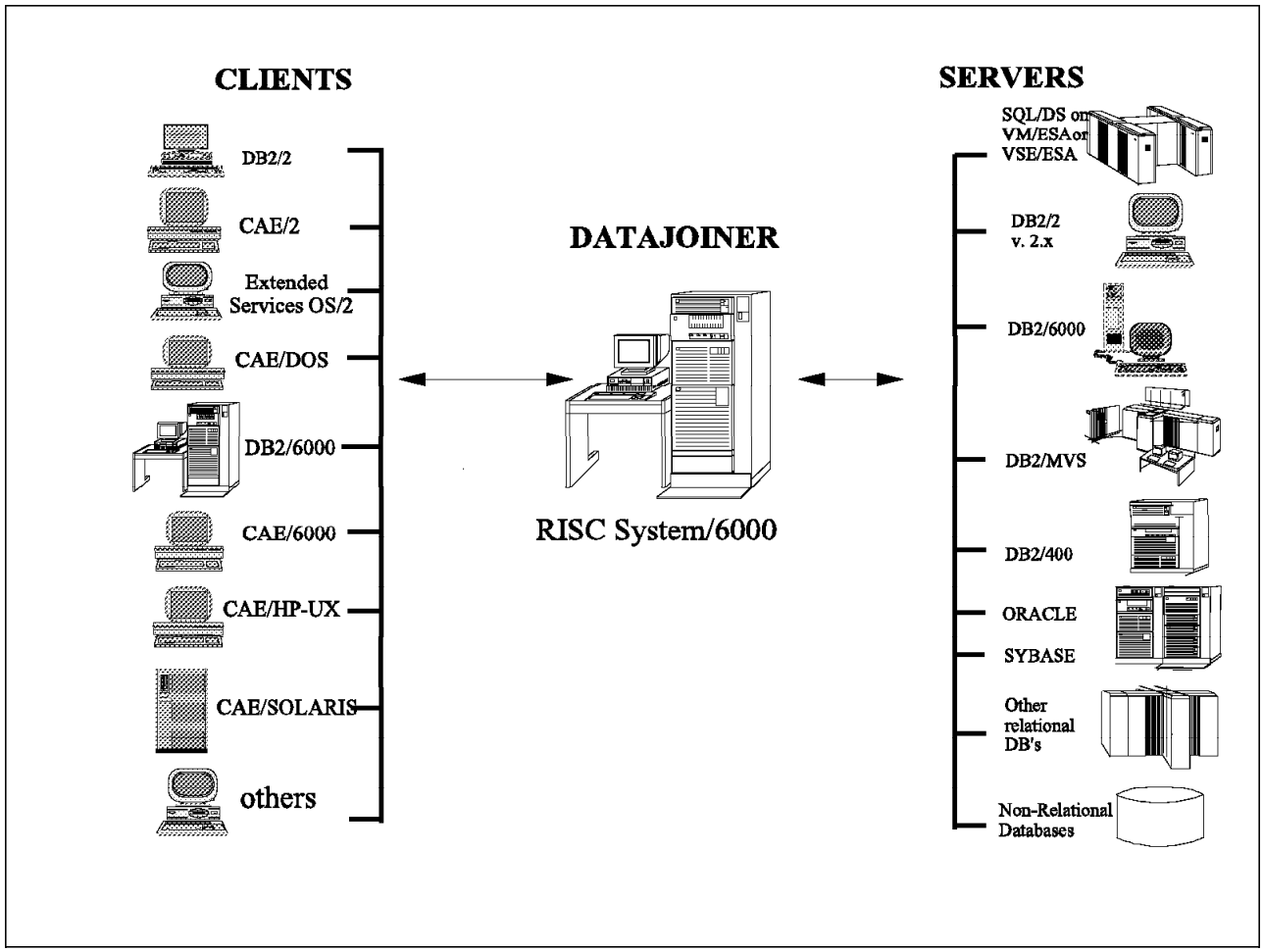


Figure 1. DataJoiner Overview

1.1.3 DataJoiner Engine Functions

The DataJoiner multidatabase system provides several key functions in support of client applications and tools. DataJoiner has the ability to perform read-only query with a distributed join of multiple remote data sources, which may reside on multiple hardware platforms, perhaps with different database technologies. DataJoiner can be used to aggregate the data from multiple sources and return the query results to a client application as if the query were answered from a

single DBMS. DataJoiner also provides a *single-site update* within a given unit of work.

DataJoiner has the ability to provide cost-based *global query optimization*. To enable the global optimizer as well as provide transparent access to enterprise data resources, DataJoiner builds and maintains a global catalog of data resources. The global catalog stores the physical location of the data and implements a table *nickname* concept to allow client applications to transparently access remote data sources using the nickname. DataJoiner automatically gathers table metadata such as column names, data types, and index information. It also keeps the global catalog refreshed based on a user-specified interval. The optimization information stored in the global catalog includes relative central processing unit (CPU) speeds, relative input/output (I/O) rates, communication bandwidth between DataJoiner and a remote data source, cardinality of the remote data table, and other information.

DataJoiner also provides error message handling to facilitate problem determination and problem resolution. DataJoiner attempts to map the error codes which it receives from remote data sources to a consistent set, so that the users′ image of being connected to a single DBMS is maintained. If the code cannot be mapped, it is forwarded to the client application “as is.”

DataJoiner uses the existing security mechanisms of the various remote data sources. Client applications must be authorized to access DataJoiner. DataJoiner stores the log-on information required to access the remote data sources. Using this information DataJoiner logs onto the remote data source on behalf of the application.

1.2 Environment Description

Figure 2 on page 5 shows the scope of the environment used in preparing this book.

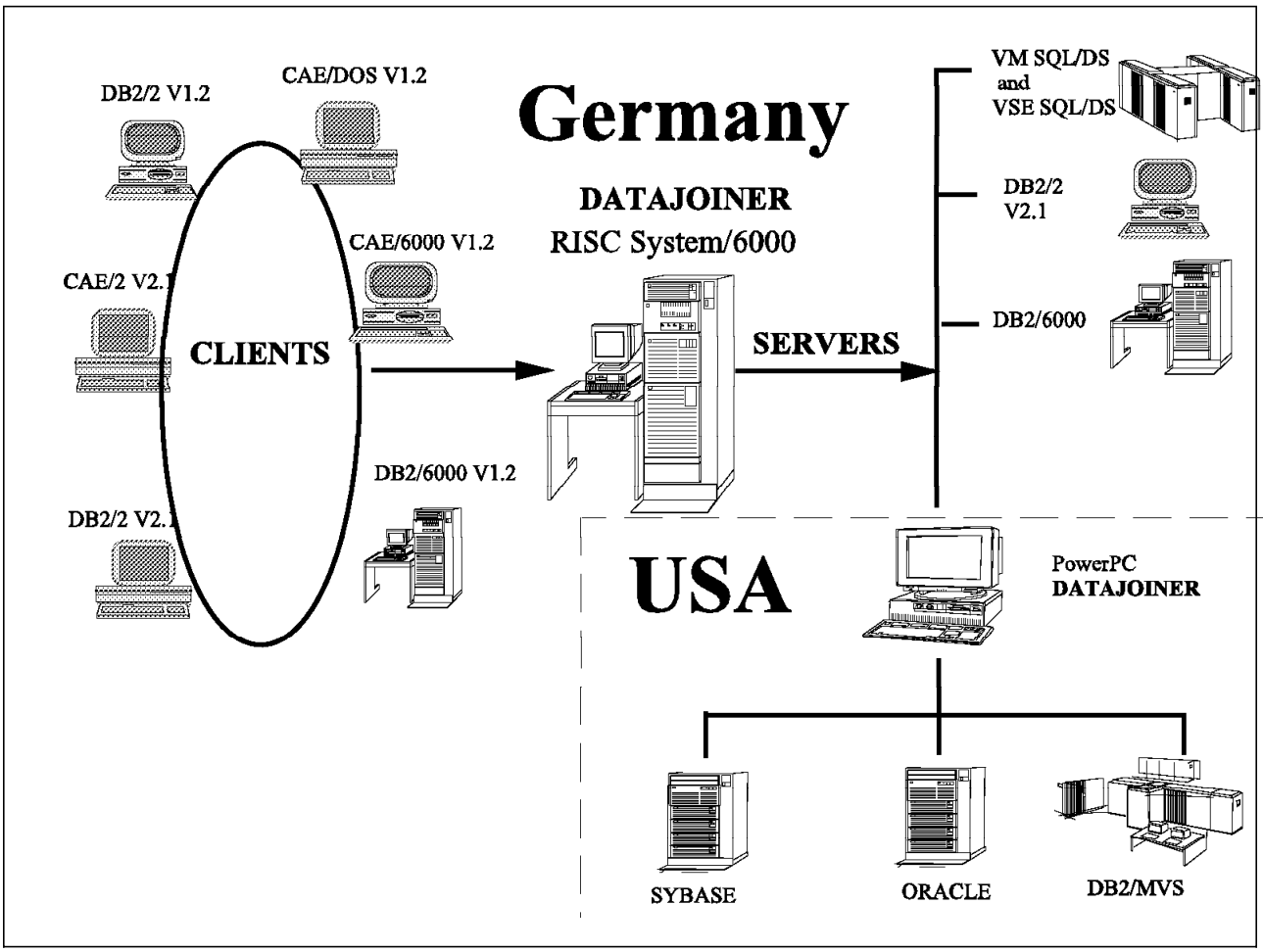


Figure 2. DataJoiner Environment

Our environment for the preparation of this book was divided into two major segments.

In Germany, we had a RISC System/6000 running DataJoiner and DB2/6000 Version 1. Clients accessing DataJoiner consisted of:

- Client Application Enabler/DOS Version 1.2
- Client Application Enabler/2 Version 2.1
- Client Application Enabler/6000 Version 1.2
- Database/2 for OS/2 Version 1.2
- Database/2 for OS/2 Version 2.1
- Database/2 for AIX Version 1.2.

All clients were installed on the same Token Ring LAN as the DataJoiner machine.

DataJoiner in Germany had direct access to:

- DB2/VM
- DB2/VSE
- DB2/6000 on a separate RISC System/6000
- DB2/6000 on the same RISC System/6000
- Another DataJoiner in the USA on a PowerPC.

In turn, DataJoiner in the USA had access to:

- DB2/MVS
- DB2/2 Version 2
- Oracle
- Sybase.

Various combinations of APPC and TCP/IP communications were used. These are described in 3.2, “Overview” on page 29.

Chapter 2. Planning and Installation

The first part of this chapter gives you an overview and information for planning your installation. The second part shows you how to install DataJoiner. After completing this chapter, you should be able to install DataJoiner and test the installation.

The installation requirements described in this chapter are based on AIX Version 3.2.5. DataJoiner also supports AIX Version 4.1.

2.1 Environmental Considerations

This section presents information that is necessary for your planning. The specific hardware and software requirements for DataJoiner are described.

2.1.1 Hardware Requirements

2.1.1.1 General Requirements

The minimum hardware configuration of DataJoiner requires a RISC System/6000 which is able to read the delivered software media. For example:

- Compact disk read-only memory (CD-ROM), or
- 8 mm tape, or
- 1/4-inch tape, or
- Other devices.

The system must also support the required version of AIX.

A minimum of about 37 MB of fixed disk storage is required (this is for DataJoiner code without remote client support). You need a minimum of 14 MB for the database that is required to work with DataJoiner. These requirements are in addition to the storage required for the base operating system and existing applications.

2.1.1.2 Requirements for Specific Configurations

The hardware requirements for your specific environment depend on the configuration you are using and how you are using your system. The necessary requirements for the operating system, other applications, application development tools, and communication products are not included in the list; use the documentation for the specific product. The disk space and memory listed in Table 1 are averages, the actual amounts depend on the functions you are using.

Table 1 (Page 1 of 2). Memory and Disk Requirements for DataJoiner

Functions	Recommended Working Set Memory (KB)	Recommended Disk (KB)
Requirements for Product Files[a]		
DataJoiner Base (djx_01_01_0000.djx)	N/A	37160
DRDA AR	N/A	1000

© Copyright IBM Corp. 1995

Table 1 (Page 2 of 2). Memory and Disk Requirements for DataJoiner		
Functions	**Recommended Working Set Memory (KB)**	**Recommended Disk (KB)**
SNA Clients	N/A	200
DBA Tool	N/A	1000
DBA Tool help/messages per locale	N/A	900
Messages per locale	N/A	950
Requirements for Each Instance of DataJoiner		
Instance Base[b]	buffer[c] + 2400	600[d]
For each additional concurrent database	buffer[c] + 1400	800[d]
DataJoiner Instance Requirements for Each Client Connection		
For each local concurrent user or application	900	N/A
For each data source accessed	150	N/A
For each additional remote client (TCP/IP)	600	N/A
For each additional remote client (APPC)	630	N/A
TOTAL (for your requirements)		

Notes:

a 15000 KB of additional temporary space is required during installation processing.

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 the user's workload. The buffer pool default is 4 MB. If you decide later that the buffer size needs to change, see the *IBM DATABASE 2 AIX/6000 Administration Guide* for instructions on how to change it. The buffer pool applies to locally stored DataJoiner data only, not data retrieved from data sources. Keep this in mind when you read instructions in the *IBM DATABASE 2 AIX/6000 Administration Guide*.

Working set memory includes the parts of a program's executable code and/or data areas that are being used intensively. Working set memory figures assume that default values are 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. It is subject to many variables. Actual requirements might differ.

The default configuration allocates 3 log files, each 4 MB in size. These logs can be relocated to another filesystem, and can be made larger or smaller as required. See the *IBM DATABASE 2 AIX/6000 Administration Guide* for more information.

To allow DataJoiner to communicate with clients and data sources, at least one network adapter is required; for example, Token Ring, Ethernet, or Integrated Services Digital Network (ISDN).

We also recommend the use of a CD-ROM for software installation and access to the product manuals.

2.1.2 Software Requirements

To run DataJoiner on RISC System/6000 you need one of the operating system releases listed below. To connect remote clients to DataJoiner, or to connect DataJoiner to remote data sources, you need either TCP/IP or APPC, or both.

2.1.2.1 Operating System

DataJoiner requires one of the following:

- IBM AIX Version 3.2 with program temporary fixes (PTFs) U403173,U412397 and U412815
- IBM AIX Version 3.2.4 or higher
  (Version 3.2.5 appears to work best.)
- IBM AIX Version 4.1

2.1.2.2 For Communication Using TCP/IP

The following software must be installed:

- djx_01_01_0000.djx
- TCP/IP (Version 3.2 or higher) component of AIX.

  To check if TCP/IP is installed, enter: `lslpp -l bosnet.tcpip*`

  If TCP/IP is installed, the output of the `lslpp -l` command would be as shown in Figure 3.

```
  Name                  State         Description
  --------------------  ------------  ---------------------------------------
Path: /usr/lib/objrepos
  bosnet.tcpip.obj 03.02.00.00
                        COMMITTED     TCPIP Applications

Path: /etc/objrepos
  bosnet.tcpip.obj 03.02.00.00
                        COMMITTED     TCPIP Applications
```

Figure 3. Output of the lslpp -1 Command if TCP/IP is Installed Correctly

2.1.2.3 For Communication Using APPC

The following software must be installed:

- djx_01_01_0000.sna_clients
- IBM AIX System Network Architecture Server/6000 (5765-247) at level 01.03.0095.0170. This is SNA Server/6000 Version 2.1. You must also apply PTF U437491.

  To check if SNA is installed at the right level, enter: `lslpp -h sna.sna.*`

  If SNA is installed at the right level, the output of the `lslpp -h` command would be as shown in Figure 4 on page 10.

```
  Name
  --------------------
    Fix Id  Release         Status     Action     Date       Time     User Name
    ------- --------------- ---------- ---------- ---------- -------- ----------
Path: /usr/lib/objrepos
  sna.sna.obj
    U427073 01.03.0093.0495 COMPLETE   APPLY      05/25/95   10:16:22 root

  sna.sna.obj
            01.03.0094.0231 COMPLETE   APPLY      05/24/95   23:07:14 root
    U432009 01.03.0094.0231 COMPLETE   APPLY      05/25/95   10:15:55 root

  sna.sna.obj
    U435033 01.03.0094.0450 COMPLETE   APPLY      05/25/95   10:11:11 root

  sna.sna.obj
    U437491 01.03.0095.0170 COMPLETE   APPLY      05/30/95   10:01:39 root

Path: /etc/objrepos
  sna.sna.obj
    U427073 01.03.0093.0495 COMPLETE   APPLY      05/25/95   10:16:25 root

  sna.sna.obj
            01.03.0094.0231 COMPLETE   APPLY      05/25/95   10:10:56 root
    U432009 01.03.0094.0231 COMPLETE   APPLY      05/25/95   10:15:58 root

  sna.sna.obj
    U435033 01.03.0094.0450 COMPLETE   APPLY      05/25/95   10:15:32 root

  sna.sna.obj
    U437491 01.03.0095.0170 COMPLETE   APPLY      05/30/95   10:04:52 root
```

Figure 4. Output of the lslpp-h Command if SNA is Installed Correctly

2.1.2.4 For Communication Using TCP/IP and APPC

The following software must be installed:

- djx_01_01_0000.sna_clients
- TCP/IP (Version 3.2 or higher) component of AIX
- IBM AIX System Network Architecture Server/6000 (5765-247) at level 01.03.0095.0170. You must also apply PTF U437491.

See Figure 3 on page 9 and Figure 4 showing examples of `lslpp` commands if TCP/IP and SNA are correctly installed.

2.1.2.5 For DataJoiner Clients

- **Local Clients**

  Normally, local clients require no communication configuration.

  X-Stations can also work as local clients. Only for this type of local client is a communication configuration required—an *X-Station configuration*. The following software must be installed to work with X-Stations:

  - TCP/IP (Version 3.2 or higher) component of AIX
  - AIXwindows Environment/6000
  - AIX Xstation Manager/6000.

- **AIX Clients**

Remote AIX clients can use either TCP/IP or APPC, or both for communication with the server.

For the required communication protocols, you can choose one or both of the following:

- TCP/IP (Version 3.2 or higher) component of AIX, or
- IBM SNA Services/6000 or IBM SNA Server/6000.

Either CAE/6000 or DB2/6000 is also required.

- **OS/2 Clients**

  Remote OS/2 clients can either use TCP/IP or APPC.

  For TCP/IP the following software is required:

  - TCP/IP for OS/2
  - CAE/2 or SDK/2.

  For APPC the following software is required:

  - CAE/2 or DB2/2
  - Communication Manager for OS/2 (CM/2).

- **DOS Clients**

  DOS clients can only work with TCP/IP. The following software is required for these clients:

  - TCP/IP for DOS
  - CAE/DOS or SDK/DOS.

- **DOS/Windows Clients**

  The requirements for DOS/Windows are the same as for DOS only.

- **Other Operating Systems**

  If you are using other operating systems you need the following functions:

  - A communication program that understands TCP/IP or APPC
  - A database program that is able to communicate with DB2/6000 using TCP/IP or APPC. This is normally a database program or a CAE program.

    Currently, there are versions of DB2 for HP-UX and Sun Solaris.

2.1.3 Supported Clients and Servers

DataJoiner supports several types of clients and servers.

All connections from a client or to a server of DataJoiner are implemented with TCP/IP or APPC. If you have a client or a server that requires another protocol, you need to use a gateway that translates the different protocols. For example, the NOV*IX Netware Server makes the translation between IPX/SPX and TCP/IP.

Not every combination of client, server, and protocol is valid. See Figure 17 on page 30 for some valid combinations.

2.1.3.1 Clients

The following list shows operating systems and the client programs that can be used on each.

Operating System	***Application***
AIX	DB2/6000
	CAE/6000
OS/2	DB2/2
	CAE/2
DOS	CAE/DOS
DOS-Windows	CAE/DOS
Win-OS/2	CAE/DOS
HP-UX	CAE/HP-UX
Solaris	CAE/Solaris

2.1.3.2 Data Sources

The following list shows the data sources that can be accessed by DataJoiner.

- DataJoiner
- DB2 for MVS
- SQL/DS for VM and VSE
- IMS
- Virtual Storage Access Method (VSAM)
- DB2/400 for OS/400
- DB2/2 V2 for OS/2
- DB2/6000 for AIX
- DB2 for HP-UX V1
- DB2 for the Solaris Operating Environment V1
- DB2 Parallel Edition (DB2/PE)
- Oracle V7.0.13
- Sybase V4R6
- Microsoft SQL Server V4.2.1
- ODBC, X/Open compliant servers
- Nonrelational database servers

Refer to the *DataJoiner Planning, Installation, and Configuration Guide* for specific maintenance levels required.

Figure 5 on page 13 shows the DataJoiner access modules that can be used to access the data sources.

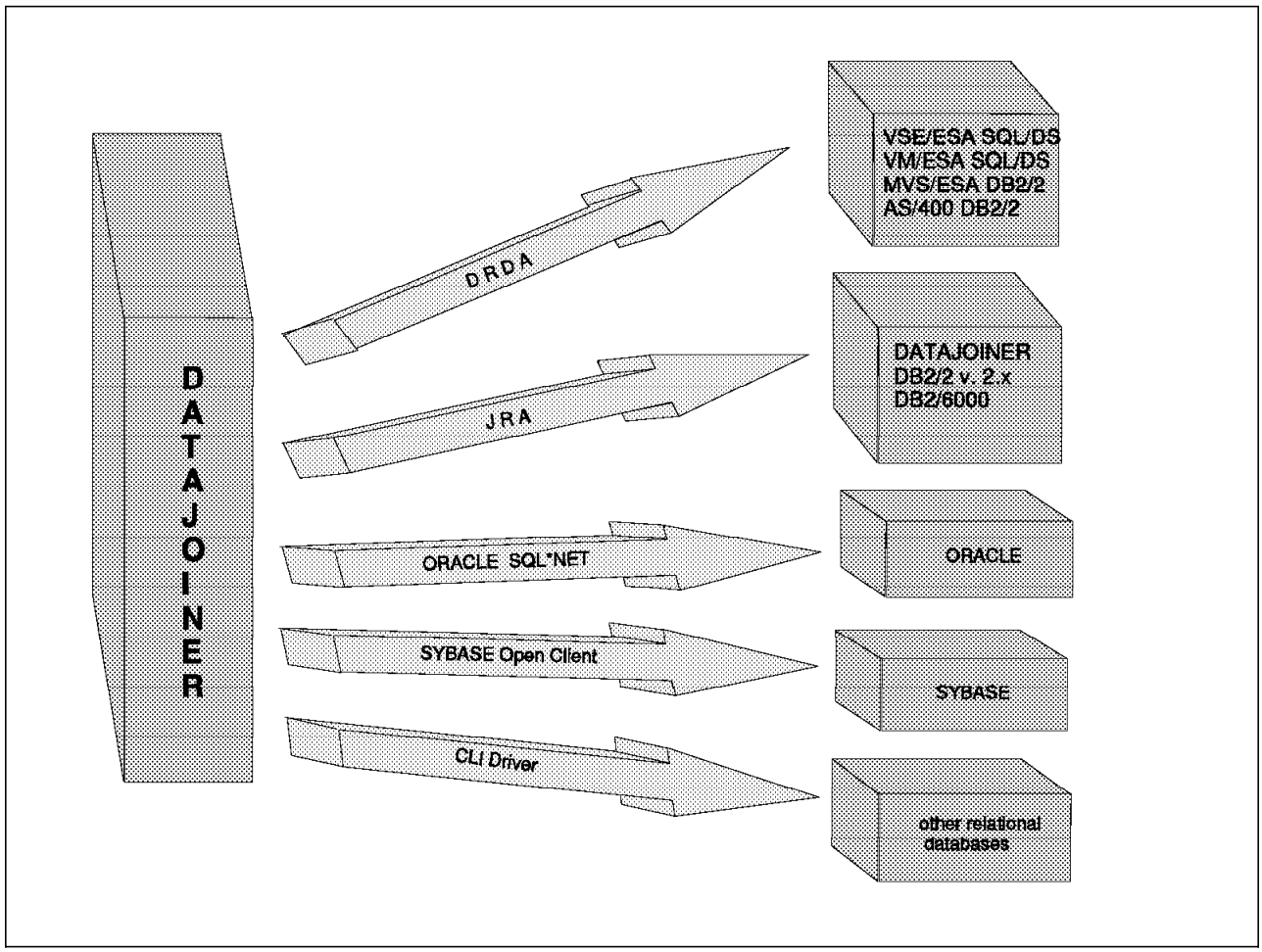


Figure 5. DataJoiner Access Modules

2.1.4 Administration Tasks

This section describes the authority levels required for tasks related to installation, administration, and operation of DataJoiner. Chapter 6, “Security” on page 145 has more detail on DataJoiner security requirements.

2.1.4.1 Root Authority

You need AIX root authority for the following tasks:

- Installing the DataJoiner code (see 2.3.1, “Installing DataJoiner” on page 17)
- Creating a group (see 2.3.3.1, “Creating a DataJoiner Instance Group” on page 19)
- Creating a user ID (see 2.3.3.2, “Creating a DataJoiner Instance User” on page 20)
- Creating an instance (see 2.3.3.3, “Creating the Instance” on page 21)
- Creating links between libraries (see 2.3.3.3, “Creating the Instance” on page 21)
- Starting DataJoiner during system boot (see 2.3.5, “Starting DataJoiner at System IPL” on page 23)
- Encryption (see 2.3.6, “Encryption” on page 24)

- Defining the database administrator (DBA) utility font (see 2.3.7, “Defining DBA Utility Font to AIX-Windows (optional)” on page 24)
- Removing the installation (see 2.4.4, “Deinstalling the DataJoiner Software” on page 27).

2.1.4.2 Instance-Owner Authority

You need instance-owner authority for the following tasks: (see 2.2, “Special Configurations,” and especially Figure 6 on page 15 , for a description of a DataJoiner instance.)

- Testing the instance (see 2.3.3.5, “Testing the Instance” on page 22)
- Creating a database (see 3.3.1, “Generating a DataJoiner Database” on page 32)
- Changing system tables (see 7.3, “Steps to Enable Users for Use of DataJoiner” on page 164)
- Removing a database (See 3.3.4, “Removing a DataJoiner Database” on page 44).

2.1.5 Security

Security requirements must be considered relatively early in the planning process. When the DataJoiner database is created, one of the parameters that must be specified is *authentication*. The value specified is very important to the overall security plan.

Many questions must be answered before security is implemented, such as these:

- Does the remote data source support trusted clients?
- If so, do the clients have facilities to support authentication?
- Will it be necessary to define user IDs and passwords to AIX on the DataJoiner server?
- How will access to DataJoiner resources be controlled?

These questions fall into two basic categories. The first is authentication and is addressed in 6.1, “Overview” on page 145. The second relates to database privileges and is covered in 7.2.2, “ Security” on page 163. Please refer to these sections for more detailed information about planning for security.

2.2 Special Configurations

You can implement your DataJoiner installation in many different ways. On one AIX system, you can have one or more *instances* of DataJoiner. A DataJoiner instance is very similar to an instance of DB2/6000 and somewhat analogous to a DB2/MVS *subsystem*; it is a single DataJoiner server described by one DataJoiner configuration file. A DataJoiner instance can include one or more databases, each of which contains a relational catalog and objects (such as tables) identified by and described in the catalog. Some examples of DataJoiner configurations are:

- Single DataJoiner instance with one database
- Single DataJoiner instance with multiple databases
- Single DataJoiner instance and single DB2/6000 instance

- Single DataJoiner instance and multiple DB2/6000 instances
- Multiple DataJoiner instances on the same machine
- Multiple DataJoiner instances on different machines
- DataJoiner and other applications and a single DB2/6000 instance.

2.2.1 General

The relationship between the DataJoiner code, an instance, a database and a table is shown in Figure 6.

You can create more than one instance based on the DataJoiner code you have installed. Under each instance, you can create more than one database, but each database can belong to only one instance. The tables you define are stored under a database. Using standard DB2/6000 Version 1 capabilities, you can work with only one database at a time.

This relationship is the same as in DB2/6000 Version 1.

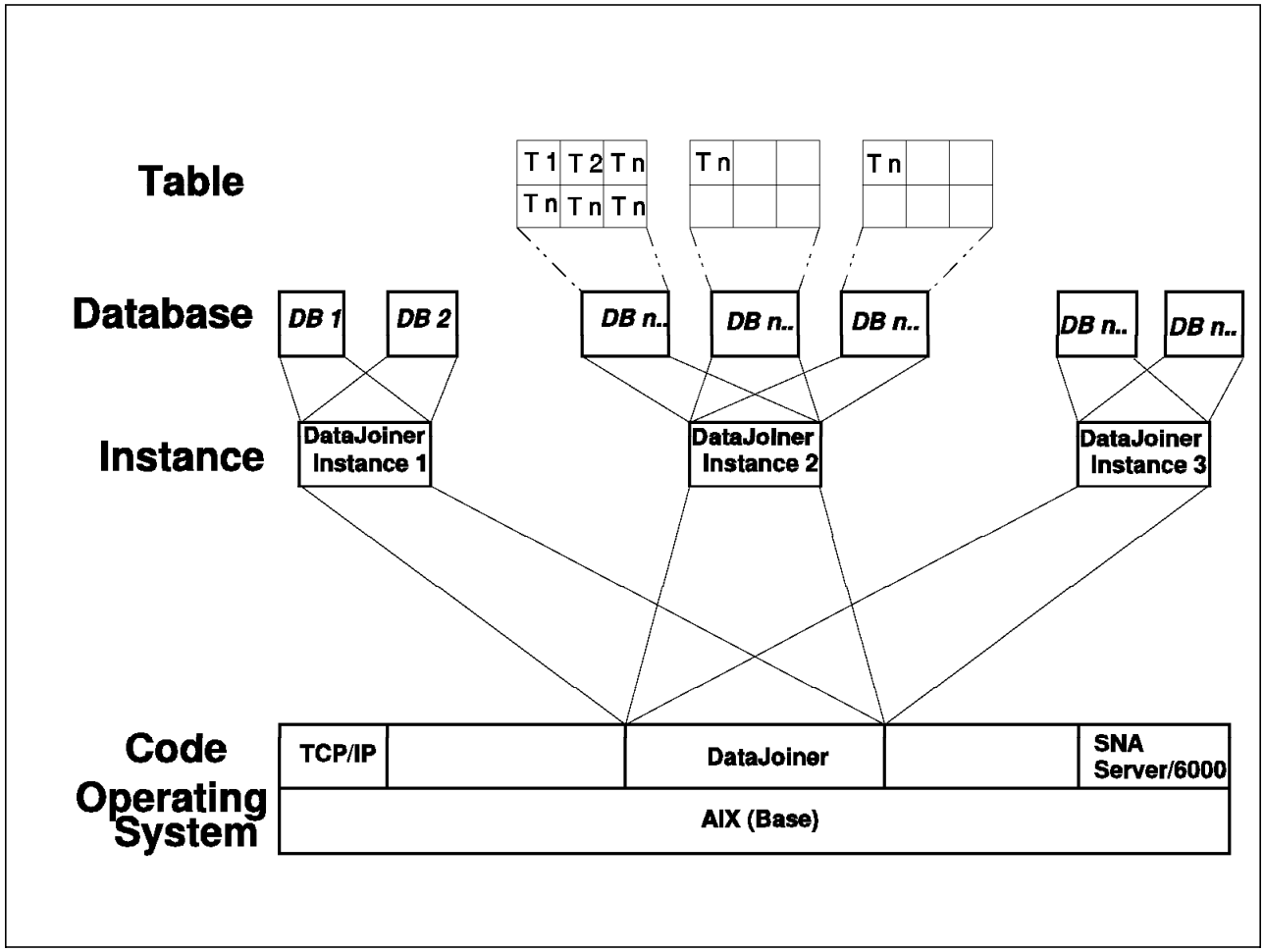


Figure 6. Relationship of Instances and Databases (Overview)

2.2.2 Using Local Clients Only

All DataJoiner users that run on the same processor as DataJoiner are local clients, even if they are using an X-station.

For local clients only, the DB2COMM variable is set to NONE, or null and no communication is defined to this instance. (DB2COMM is an environment variable. The statement that sets it is in the profile shell script that is created when you create a DataJoiner instance.)

2.2.3 Using Remote TCP/IP Clients

All DataJoiner users that run on a different machine and communicate with the TCP/IP protocol from their CAE program or database program to DataJoiner are called remote TCP/IP clients.

In ths case, the DB2COMM variable is set to TCPIP and the TCP/IP communication is defined to this instance. Alternatively, set DB2COMM to null, in which case the communication support is determined by the `service_name` and `tpname` keywords in the database manager configuration file as described in *DataJoiner Planning, Installation, and Configuration Guide*.

2.2.4 Using Remote APPC Clients

All DataJoiner users that run on a different machine and communicate with the SNA (APPC) protocol from their CAE program or database program to DataJoiner are called remote APPC clients.

In this case, the DB2COMM variable is set to APPC and the SNA (APPC) communication is defined to this instance. Alternatively, set DB2COMM to null, in which case the communication support is determined by the `service_name` and `tpname` keywords in the database manager configuration file as described in *DataJoiner Planning, Installation, and Configuration Guide*.

2.2.5 Using Remote TCP/IP and APPC Clients

All DataJoiner users that run on a different machine and communicate with the TCP/IP protocol, the SNA (APPC) protocol, or both, from their CAE program or database program to DataJoiner are called remote TCP/IP or APPC clients.

The DB2COMM variable is set to TCPIP, APPC, or null, and the TCP/IP and the SNA (APPC) communication is defined to this instance.

2.3 Installation of Software

The following procedure can be used to install the DataJoiner product.

The main installation steps are:

1. Use System Management Interface Tool (SMIT) or `installp` command to install DataJoiner on your system.
2. Verify the installation (optional).
3. Create an instance of the product.
4. Create links between system libraries and database libraries (optional: not recommended for production environment).
5. Set environment variables for the instance and local clients (AIX user IDs).
6. Start DataJoiner during system initial program load (IPL) (optional).

2.3.1 Installing DataJoiner

There are two ways to install the DataJoiner software: use SMIT or use the command line.

2.3.1.1 Installation of DataJoiner Using SMIT

1. Login as user root.
2. Place installation media in device (tape, CD, diskette).
3. Type `smit install_latest`.
4. Press F4 to select the installation device (tape, CD, diskette).
5. Press F4 and select the products you wish to install using F7.
6. After selecting products, press Enter.

```
                Install Software Products at Latest Available Level


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

                                                      [Entry Fields]
* INPUT device / directory for software               /dev/rmt0.1
* SOFTWARE to install                                 [1.1.0.0  djx_01_0_000>  +
  Automatically install PREREQUISITE software?        yes                      +
  COMMIT software?                                    no                       +
  SAVE replaced files?                                yes                      +
  VERIFY software?                                    no                       +
  EXTEND file systems if space needed?                yes                      +
  REMOVE input file after installation?               no                       +
  OVERWRITE existing version?                         no                       +
  ALTERNATE save directory                            []


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

Figure 7. SMIT Installation Screen

2.3.1.2 Installation of DataJoiner from the Command Line

1. Login as user root.
2. Place installation medium in device (tape, CD, diskette).
3. Use the `installp` command to install the software and the necessary prerequisites. Its format is:

   ```
   installp -qgaXd <device> <product>
   ```

   where

   - <device> is the installation medium
   - <product> is the component of DataJoiner you want to install

     An example of the command is

     ```
     installp -qgaXd /dev/rmt0 djx_01_01_0000.djx
     ```

     which installs and applies DataJoiner plus all dependent software (without client support) from the tape device.

List of Flags Used with the installp Command

Flag	Description
-a	Apply software
-c	Commit software
-d	Install device
-g	Include all prerequisite software
-q	Quietmode (do not prompt for device to be mounted)
-X	Expand filesystem if required
-F	Overwrite existing level of software even if existing level is newer

2.3.2 Verifying Installation of Software

To check that the software has installed without error, execute the following commands:

- `lppchk -c <product>`
- `lppchk -v <product>`

where

- <product> is the component of DataJoiner you want to install.

  Examples of these commands are

  - `lppchk -c djx_01_01_0000.djx`
  - `lppchk -v djx_01_01_0000.djx.`

  The command returns with a nonzero code and issues messages if errors are found. Otherwise, it returns with a code of zero.

List of Flags Used with the lppchk Command

Flag	Description
-c	Verifies the checksum and verifies that the file sizes are consistent with the Software Vital Product Data (SWVPD) database information.

-f	Verifies that the file size is consistent with the SWVPD.
-l	Verifies symbolic links for files as specified in the SWVPD.
-u	Updates the SWVPD with new checksum or size information from the system when the system information does not match the SWVPD database. This flag is valid only with the -c or -i flag.
-v	Verifies that the / (root) , /usr and /usr/share parts of the system are valid with each other.

2.3.3 Creating a DataJoiner Instance

You can have more than one instance of DataJoiner. Every DataJoiner instance is defined by its instance owner. The login name of the instance owner is also the name of the DataJoiner instance. The instance owner must be unique for each DataJoiner instance.

To create an instance of DataJoiner, complete the following steps:

1. Create a DataJoiner system administrative group (instance group).
2. Create a user (instance owner).
3. Create the instance.
4. Set environmental variables.

These steps are described in detail below.

2.3.3.1 Creating a DataJoiner Instance Group

Create a group within AIX. This will be the system administrative group for the instance. This group can also be used to allow other users to have system administration (sysadm) authority for the instance. However, the group does not have to be an administrative group within AIX. This is an AIX operating system function and is not required for administration within DataJoiner.

All users that belong to this group have sysadm authority for this instance.

To create the group:

- From the command line:
  - Login as user root.
  - Enter `mkgroup -A djadm1`, which creates a group called djadm1.
- From SMIT:
  - Login as user root.
  - Enter `smitty mkgroup`. Type the group name in the required field (see Figure 8 on page 20).

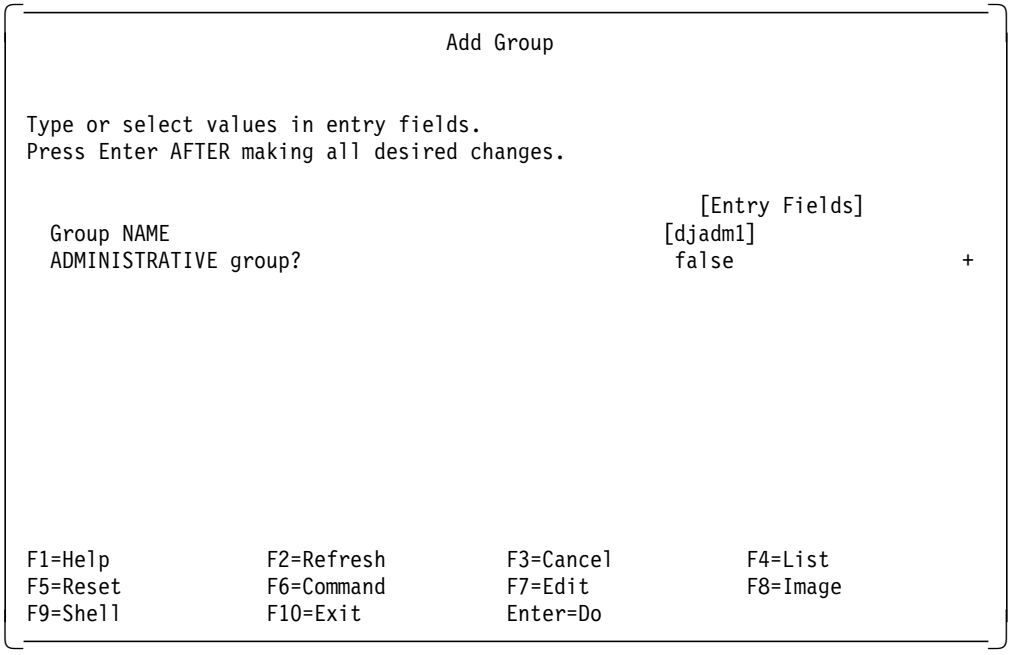


Figure 8. SMIT Add a Group

2.3.3.2 Creating a DataJoiner Instance User

Create a user who will be the instance owner. The login name used here will also be the name given to the instance owner. The instance owner's primary group must be the system administrative group you created in the previous step. In our example, the group was djadm1.

To create the user:

- From the command line:
  - Login as user root.
  - Enter:

    ```
    mkuser pgrp=djadm1 djinst1
    ```

    This will create a user called djinst1 whose primary group is djadm1. User djinst1 also belongs to the group called *staff* (by default if you didn't change the definitions in /etc/security/mkuser.default). The home directory will be set to the default value, /u/djinst1 or /home/djinst1, depending on your default settings in the /etc/security/mkuser.default file.
- From SMIT:
  - Login as user root.
  - Enter `smitty mkuser` and type in the required information (see Figure 9 on page 21).

```
                                  Create User

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

[TOP]                                                  [Entry Fields]
* User NAME                                           [djinst1]
  ADMINISTRATIVE User?                                  false                  +
  User ID                                              []                      #
  LOGIN user?                                           true                   +
  PRIMARY group                                        [djadm1]                +
  Group SET                                            []                      +
  ADMINISTRATIVE groups                                []                      +
  SU groups                                            [ALL]                   +
  HOME directory                                       []
  Initial PROGRAM                                      []
  User INFORMATION                                     []
  Another user can SU to user?                          true                   +
  User can RLOGIN?                                      true                   +
  TRUSTED PATH?                                         nosak                  +
[MORE...12]

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

Figure 9. SMIT Create User

Any other user who belongs to the system administrative group for this instance (here, djadm1) will have sysadm authority for this instance. However the system administrative group does not have to be their primary group. This condition is a requirement for the instance owner only.

- Assign a password to the user you have just created. Use SMIT or from the command line enter passwd <userid>, where
  - <userid> is the name of the user you just created.

    For example,

    passwd djinst1

2.3.3.3 Creating the Instance

Create an instance of DataJoiner. You must use the instance owner's login name to create the instance.

- Login as user root.
- Enter:

  /usr/lpp/djx_01_01_0000/instance/db2instance <userid>

  where
  - <userid> is the name of the user you just created.

    For example,

    /usr/lpp/djx_01_01_0000/instance/db2instance djinst1.

  This command creates a directory called sqllib in the home directory of the instance owner, in our example /home/djinst1/sqllib. This directory contains symbolic links to directories in /usr/lpp/djx_01_01_0000 and also copies of instance-independent files, such as db2profile and db2dump.

2.3.3.4 Setting Environment Variables

Set environment variables for the instance owner and local clients to create their environment. DataJoiner builds two script files when the instance is created . You can find these files in the sqllib directory under the instance owner's home directory. These files are db2profile (used for a Bourne or a Korn shell) and db2cshrc (used for a C shell).

You can use any of several ways to set your environment (see *DataJoiner Planning, Installation, and Configuration Guide*). The following is what we used:

For running under a Bourne or a Korn shell,

- Execute the db2profile from the user's .profile.

  Edit your .profile and add this line:

  ```
  . /home/djinst1/sqllib/db2profile
  ```

If you are running under a C shell,

- Execute the db2cshrc from the user's .login.

  Edit .login and add this line:

  ```
  . /home/djinst1/sqllib/db2cshrc
  ```

2.3.3.5 Testing the Instance

Before using DataJoiner, you must set the variables and start the DataJoiner instance. Login as the instance owner.

To see if you are using the right instance, enter the `db2 get instance` command. If you get the following message:

```
SQL10007N Message "-1390" could not be retrieved.  Reason code: "1".
```

your DB2INSTANCE variable is not set correctly. Check the DB2INSTANCE variable and the file where you define this variable to determine if DB2INSTANCE is defined correctly. Figure 10 shows an example.

```
$ db2 get instance

 The current database manager instance is:  djinst1

$
```

Figure 10. Example of db2 get instance

To start DataJoiner, enter:

```
db2start
```

This starts the DataJoiner processes associated with this instance; it must be done before any DataJoiner (DB2) command can run against this instance.

To stop the DataJoiner instance, enter:

```
db2stop
```

This stops the DataJoiner processes associated with this instance unless current connections to a database still exist.

2.3.4 Creating Links Between System and DataJoiner Libraries (Optional)

You may want to create links from your system libraries to the DataJoiner libraries. This can be useful in an application development environment. For example, links could eliminate the need to specify the full path to the product libraries and include files. We do not recommend this procedure for production systems, and it degrades the ability to coexist with DB2/6000 on the same AIX system.

The DataJoiner libraries and include files are all stored in two directories: /usr/lpp/djx_01_01_0000/lib and /usr/lpp/djx_01_01_0000/include.

If you want to create these links, use the following command:

```
/usr/lpp/djx_01_01_0000/cfg/db2ln
```

To remove these links, use the following command:

```
/usr/lpp/djx_01_01_0000/cfg/db2rmln
```

Note:

To execute these commands, you must be the user root.

2.3.5 Starting DataJoiner at System IPL

If you wish to have your instance of DataJoiner started automatically when the system is booted, you can use the following steps:

1. Login as user root.
2. Use an editor to create or edit the file named /etc/rc.db2. Add a line such as the following for each instance that is to be started at system boot:

   ```
   su - djinst1 "-c db2start >/dev/console 2>&1"
   ```

3. Set the permissions on the /etc/rc.db2 file to 744 if not already done.

   ```
   chmod 744 /etc/rc.db2
   ```

4. Add an entry in /etc/inittab file by using the following command:

   ```
   mkitab "startdb2:2:once:/etc/rc.db2 > /dev/console 2>&1"
   ```

Figure 11 on page 24 shows a listing of a /etc/rc.db2 file.

```
#!/bin/sh
#
# COMPONENT_NAME: (rc.db2) DB2(DataJoiner) rc script
#
# FUNCTIONS:
#
#
# (C) COPYRIGHT International Business Machines Corp. 1995
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.


#  Uncomment the following line to start the DataJoiner (DB2)
su - djinst1 "-c db2start >/dev/console 2>&1"
```

Figure 11. File /etc/rc.db2

2.3.6 Encryption

DataJoiner may need passwords to access data sources. If so, the passwords can be stored in the system catalog as described in 5.1.4.2, “Add an Entry to SYSIBM.SYSREMOTEUSERS” on page 98.

DataJoiner encrypts passwords when they′re added to the catalog and decrypts them just before attempting to access the data sources. Users and applications are never allowed to see passwords. However, if you feel that DataJoiner′s encryption is not sufficient for your organization, DataJoiner provides user exits that let you add encryption on top of the security measures already provided.

Please refer to Appendix A in the *DataJoiner Planning, Installation, and Configuration Guide* for information on Encryption User Exits.

2.3.7 Defining DBA Utility Font to AIX-Windows (optional)

The DataJoiner database administration (DBA) tool utility uses a special font to display icons. There are several steps that should be completed on every workstation to display the DBA tool.

If you omit this step, the following warning message appears:

```
Cannot convert string "-ibm-*-db2v1" to type FontStruct
```

when you use the `db2adm` command. The command will still work without problems.

There are two ways to make these fonts available. The first is the more efficient.

1. Install the font in the default font directory `/usr/lib/X11/fonts`.

   To do this, take the following steps:
   a. Login as user root.
   b. Copy the file
      `<instance owners home directory>/sqllib/dbat/fonts/db2v1.bdf` to
      `/usr/lib/X11/fonts` using the AIX `cp` command.
   c. Change the current directory to `/usr/lib/X11/fonts`.

d. Enter the command `bdftopcf db2v1.bdf > db2v1.pcf`
e. Enter the command `mkfontdir`

2. Update each user's .xinitrc file. Add the following line:

```
xset fp+ $HOME/sqllib/dbat/fonts
```

2.3.8 Installing DataJoiner and DB2/6000

This section provides information on installing DataJoiner on a workstation where DB2/6000 is already installed, or is being installed at the same time.

The following is a checklist for installation of DataJoiner and DB2/6000. If DB2/6000 is already installed, omit the DB2/6000 steps.

1. Create a filesystem for your DataJoiner or DataJoiner instance (optional).
2. Create a filesystem for your DB2 or DB2 instance (optional).
3. Install the DataJoiner code (see 2.3.1, "Installing DataJoiner" on page 17).
4. Install the DB2/6000 code (similar to 2.3.1, "Installing DataJoiner" on page 17).
5. Create a DataJoiner instance (see 2.3.3, "Creating a DataJoiner Instance" on page 19).
6. Create a DB2/6000 instance (similar to 2.3.3, "Creating a DataJoiner Instance" on page 19).
7. Make DataJoiner and DB2/6000 available at startup time (optional). See 2.3.5, "Starting DataJoiner at System IPL" on page 23 for more information.

   Figure 12 is a listing of the /etc/rc.db2 file for DB2 and DataJoiner. In this example, the DB2 instance is called db2inst1.

```
#!/bin/sh
#
# COMPONENT_NAME: (rc.db2) DB2/6000 and DataJoiner rc script
#
# FUNCTIONS:
#
#
# (C) COPYRIGHT International Business Machines Corp. 1995
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.


#  Uncomment the following line to start the DataJoiner (DB2)
su - djinst1 "-c db2start >/dev/console 2>&1"

#  Uncomment the following line to start the DB2/6000
su - db2inst1 "-c db2start >/dev/console 2>&1"
```

Figure 12. File /etc/rc.db2 for DB2 and DataJoiner

8. Change the /etc/services file

   This is required if you want to access DB2/6000 by using DataJoiner.

   Figure 13 on page 26 is an excerpt from an /etc/services file.

```
instsrv         1234/tcp                        # network install service
ingreslock      1524/tcp
writesrv        2401/tcp                        # temporary port number

#
# DataJoiner and/or DB2 specific services
#
djxtcpip        2455/tcp                        # DataJoiner TCP/IP
djxinter        2456/tcp                        # DataJoiner interrupt

db2tcpip        2555/tcp                        # DB2/6000 TCP/IP
db2inter        2556/tcp                        # DB2/6000 interrupt
```

Figure 13. File /etc/services for DB2 and DataJoiner

9. Synchronize the /etc/services file (see 3.3.2.3, “Synchronization of the /etc/services file” on page 36 for more information).
10. Configure the TCP/IP port in DataJoiner (see 3.3.2, “Configuring DataJoiner for TCP/IP Clients” on page 35 for more information).
11. Configure the TCP/IP port in DB2/6000.
12. Update the DataJoiner tables.

2.4 Deinstallation

This section provides information on how to deinstall or remove parts of your installation or the complete installation.

2.4.1 General Guidelines

Products can be removed only if they are in the APPLY state.

You should stop all processes that are using parts of the product you want to remove.

Do not forget to remove links (created by the `ln` command) you may have created. These links are not removed automatically; they still exist, even though they point to nothing.

2.4.2 Removing a Database

To remove a database you can use the command line or the DBA Tool. On the command line, do the following:

1. Login as the instance owner.
2. Use the `db2 drop database <database name>` command.

For more information see 3.3.4, “Removing a DataJoiner Database” on page 44.

2.4.3 Removing an Instance

Before you can remove an instance, all databases under this instance must be removed. See 2.4.2, “Removing a Database,” or for detailed information see 3.3.4, “Removing a DataJoiner Database” on page 44.

A DataJoiner instance can be removed by performing the following steps. This procedure removes only the instance; the user ID is still available.

1. Login as the instance owner.

2. Ensure that the database manager for this instance is stopped.
3. Make sure that you are not in the sqllib directory, then remove the $HOME/sqllib directory using the rm command, which removes the directory and all its files.

If you want to remove the instance and the user ID, omit steps 1 to 3 above and use the following procedure:

1. Login as user root.
2. Execute rmuser ’-p’ <userid>

   where

   <userid> is the user ID of the instance owner (instance) you want to remove.

   or

   Use SMIT to remove the user ID.
3. Remove the home directory of this user by using the rm -r command.

2.4.4 Deinstalling the DataJoiner Software

The DataJoiner code can be deinstalled only if it is in the APPLY state. If it is in the COMMIT state, it cannot be removed.

There are two ways to deinstall the DataJoiner software: use SMIT or use the command line.

Use the following procedure with SMIT:

1. Login as user root.
2. Enter smit install_remove

   You then see a screen such as Figure 14.

```
                        Remove Applied Software Products

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

                                                      [Entry Fields]
* SOFTWARE name                                     []                      +
  Automatically remove DEPENDENT software?           no                       +
  EXTEND file systems if space needed?               yes                      +









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

Figure 14. Remove Applied Software Using SMIT

3. Press F4 (the line labeled SOFTWARE name must be highlighted)

   You then see a screen such as Figure 15 on page 28.

```
                    Remove Applied Software Products


Move cursor to desired item and press F7. Use arrow keys to scroll.
    ONE OR MORE items can be selected.
Press Enter AFTER making all selections.

[TOP]
  # Name                                  Level
  #-------------------------------------------------------------------
    djx_01_01_0000.djx                   01.01.0000.0000
    djx_01_01_0000.drda_ar               01.01.0000.0000
    djx_01_01_0000.sna_clients           01.01.0000.0000
    djx_01_01_0000.xdba                  01.01.0000.0000
    djx_01_01_0000.xdbamsg.En_US.xdba    01.01.0000.0000
    sna.snapi.obj                        01.01.0093.0495
[MORE...15]

F1=Help                 F2=Refresh              F3=Cancel
F7=Select               F8=Image                F10=Exit
Enter=Do                /=Find                  n=Find Next
```

Figure 15. SMIT Remove Applied Software (Select Screen)

4. Move the cursor to the product you want to remove and press F7. Repeat this for each additional product you want to remove.
5. Press Enter when you are finished with your selection. You will then see the screen shown in Figure 14 on page 27 again.
6. Press Enter again. You will get a confirmation popup window.
7. Press Enter again; the system starts the removal process.

Chapter 3. Configuration of DataJoiner for Clients

3.1 Objectives

This chapter provides information on how to set up and configure a DataJoiner environment for use by the supported clients.

The chapter addresses the following topics:

- Overview of the environments that are supported
- Configuration of the DataJoiner server
- Configuration of DataJoiner for TCP/IP Clients
- Configuration of DataJoiner for APPC Clients.

3.2 Overview

Figure 16 illustrates an example of a configuration that shows how DataJoiner opens a new dimension in connectivity. DataJoiner provides transparent data access across systems that may have different architectures, with the capability to access many data sources at the same time.

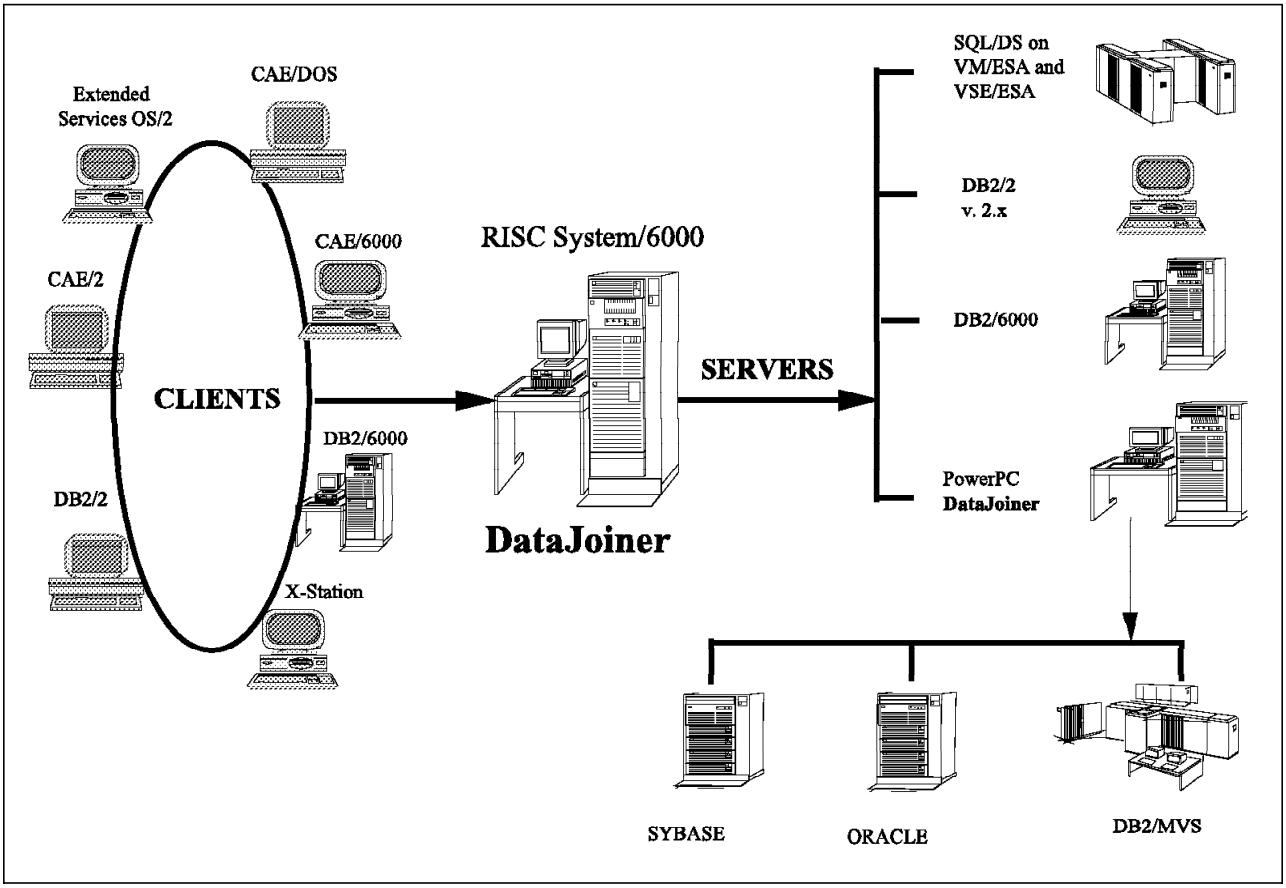


Figure 16. Connections Using DataJoiner Facilities

© Copyright IBM Corp. 1995

In the example in Figure 16, the clients are connected in a LAN. Some servers are also connected with a LAN and others with a wide area network (WAN).

These connections use different protocols. DataJoiner can receive an SQL request using one protocol and then convert it to the specific protocol for each system that is involved.

Figure 17 shows the type of protocols used from different clients to DataJoiner, and from DataJoiner to different data sources.

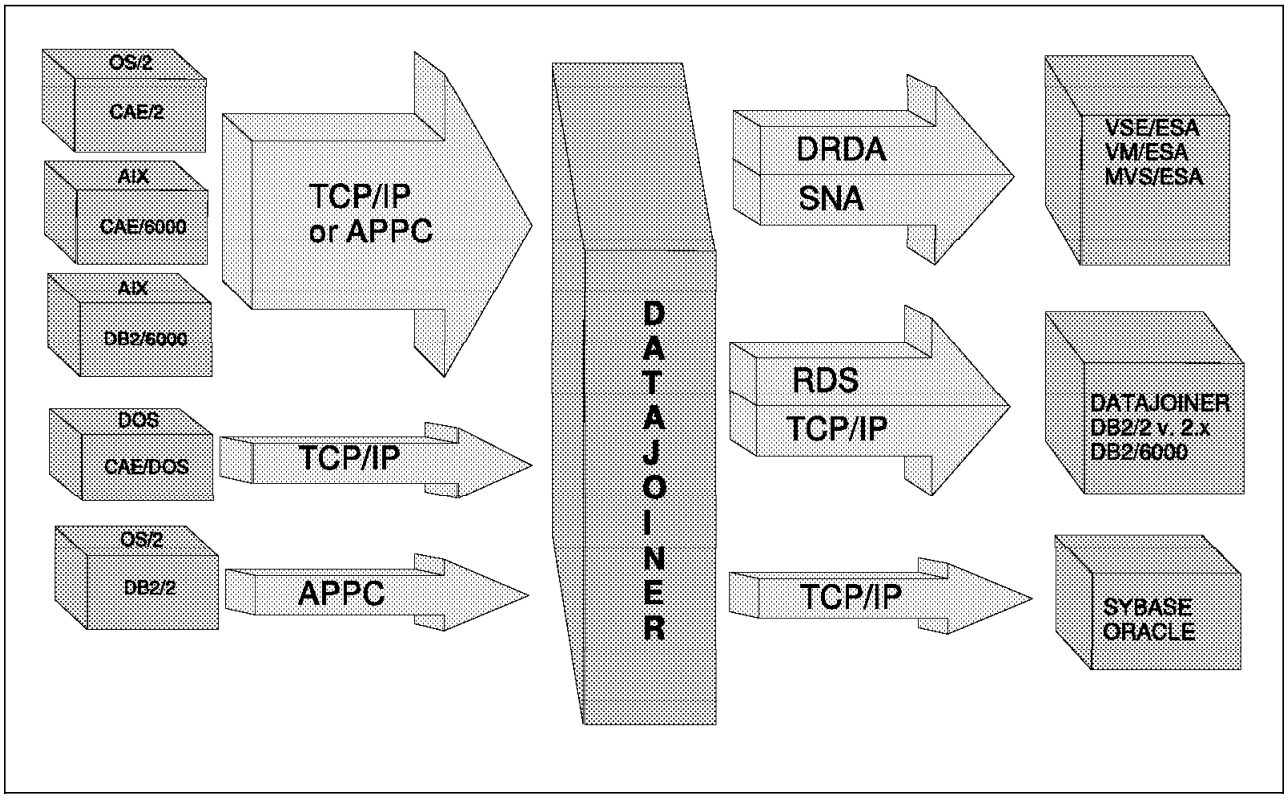


Figure 17. Protocols Used in DataJoiner Connections

The environment implemented in the ITSO for development of this book used local hosts as remote data sources and employed a WAN to communicate with other hosts. See Figure 18 on page 31. To connect to the WAN, two different types of controllers were available. The Multiprotocol Network Program (MPN 6611) supported TCP/IP as the communication protocol. Both a 3745 NCP controller attached to a VM/ESA system (Network Host) and a 3174 controller attached to an Application Host were available to support SNA as the communication protocol. We used the 3174 controller for our connections, so that the definitions we used are defined to VTAM in the Application Host.

All the configuration files are provided as Appendixes in this book. Figure 18 on page 31 also includes another VM/ESA system with a DB2/VM (SQL/DS) Version 3.4 database, and a VSE/ESA guest system under this VM/ESA. DB2/VM (SQL/DS) Version 3.4 database is also running under the VSE/ESA guest.

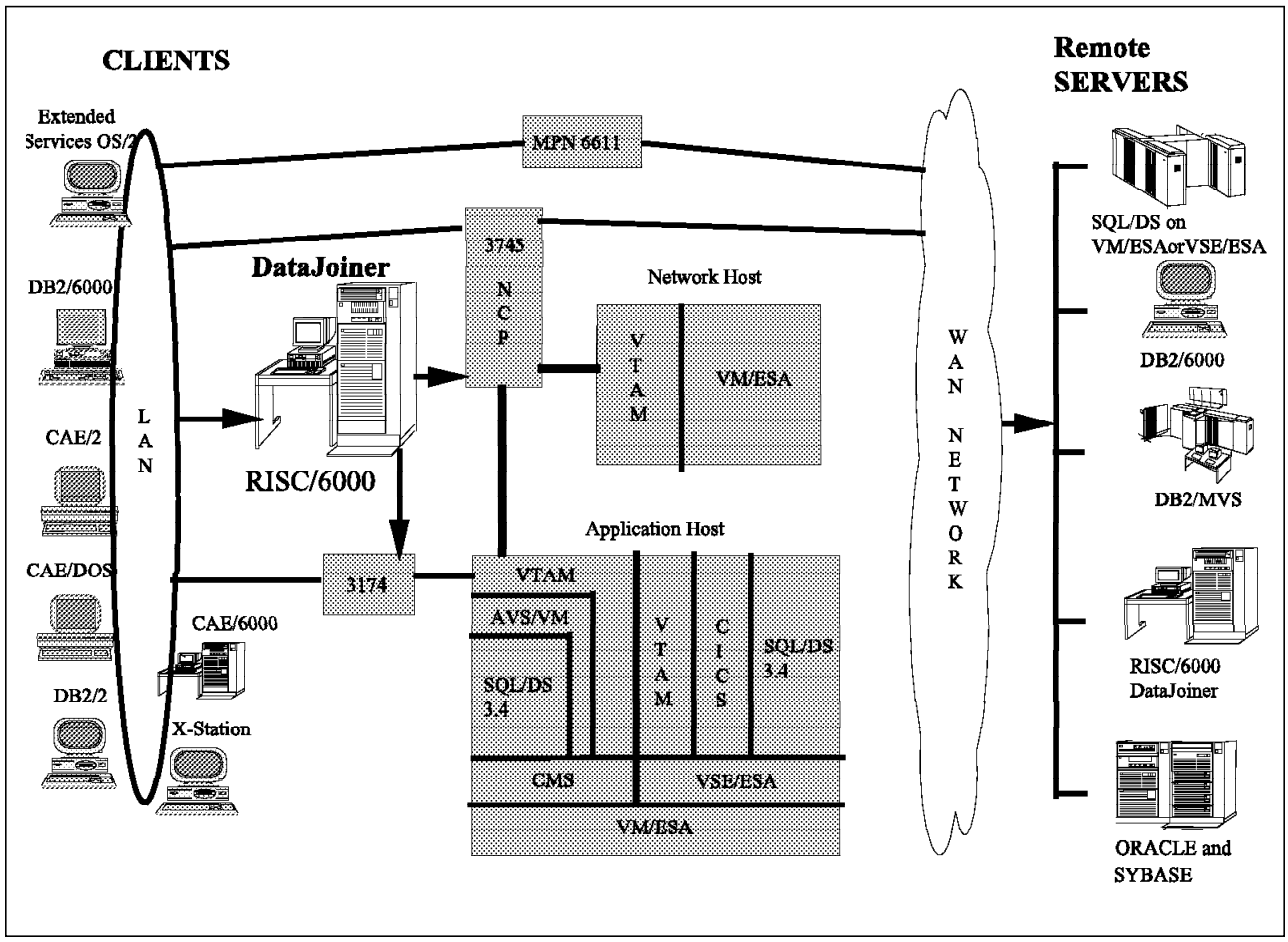


Figure 18. Connections to the Network

The next sections show the configurations for this environment.

3.3 Configuration of DataJoiner

Configuring DataJoiner requires configuring all the components involved in connectivity. DataJoiner can be connected to the clients and servers using different protocols. Clients can communicate with DataJoiner using TCP/IP or APPC. Servers can communicate with DataJoiner using TCP/IP or SNA protocol.

Figure 19 shows how the major components are related.

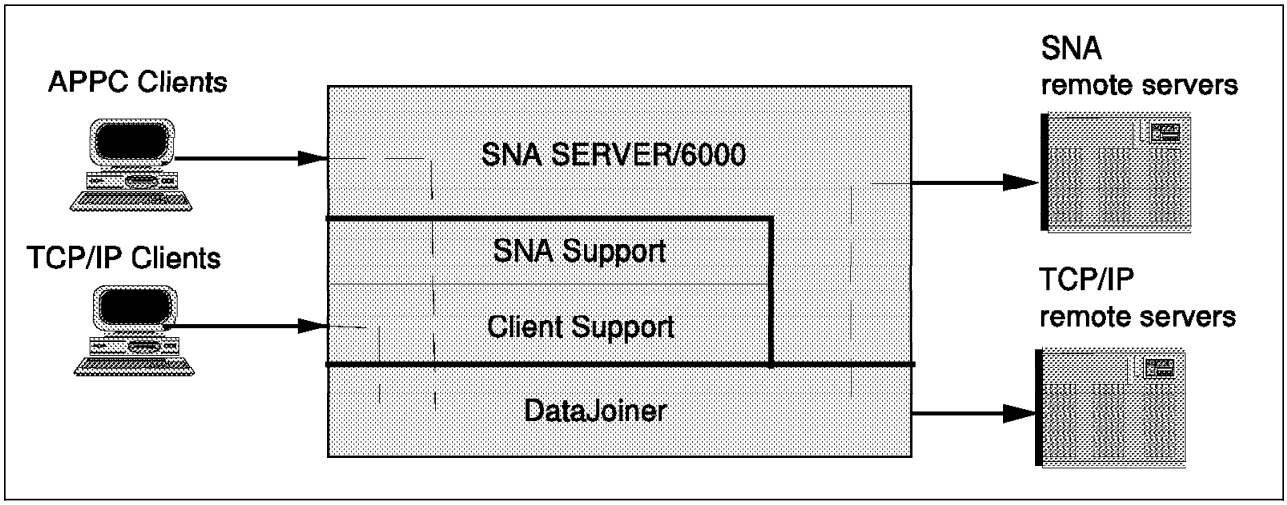


Figure 19. DataJoiner Connectivity

As two machines now run DataJoiner (one in Germany and another in the USA), each must be configured for its specific environment. See Figure 2 on page 5.

The remainder of this chapter discusses

- Generating a DataJoiner database
- Configuring DataJoiner for TCP/IP clients
- Configuring DataJoiner for APPC clients
- Removing a DataJoiner database.

3.3.1 Generating a DataJoiner Database

After installing the DataJoiner product, you must create at least one database to be able to give DataJoiner the necessary information about the remote data sources (servers) to be accessed.

To create a database you need about 14 MB of free space in the file system where the database will reside. By default, this is the home directory of the instance owner.

Only the *instance owner*, or a user with sufficient privilege, can establish this configuration.

The DataJoiner database can be created by using the *db2* Command Line Processor or the DataJoiner Administration Utility (*db2adm*).

Before you create the database, be sure that the database manager is already started. If it is not started, enter the following command:

```
db2start
```

3.3.1.1 Creating the Database by Using the Command Line

You can issue the commands in either of two different ways:

- Start the db2 command line processor (CLP) and execute all commands from there.

  or
- Execute all commands directly from the AIX shell. From the shell, you must always add db2 in front of the command.

Using the AIX shell to create the database, enter:

```
db2 create database <database name> authentication <xxxx>
```

where

- <database name> is the name of your database.
- <xxxx> is an important security option; its default value is **server** (for details see 6.2.1, “Concept of Authentication” on page 145).

Here is an example:

```
db2 create database datajoin
```

The command will take a few minutes to complete. To check that the database was created successfully, issue the command

```
db2 list database directory
```

The output should resemble Figure 20.

```
 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                 = DATAJOIN
 Database name                  = DATAJOIN
 Local database directory       = /home/djinst1
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Indirect
 Authentication                 = SERVER
```

Figure 20. List Database Directory Output

3.3.1.2 Create the Database with the Database Administration Tool

The database administration tool (DBA) is a Motif-based application.

To start this tool, enter from a terminal running AIXwindows:

```
db2adm &
```

A window is then displayed as shown in Figure 21 on page 34.

Figure 21. DBA Utility Window (Main)

To create the database, click on the Open button. On the following screen, click on the Create button. You then see the **Create Database** window as shown in Figure 22.

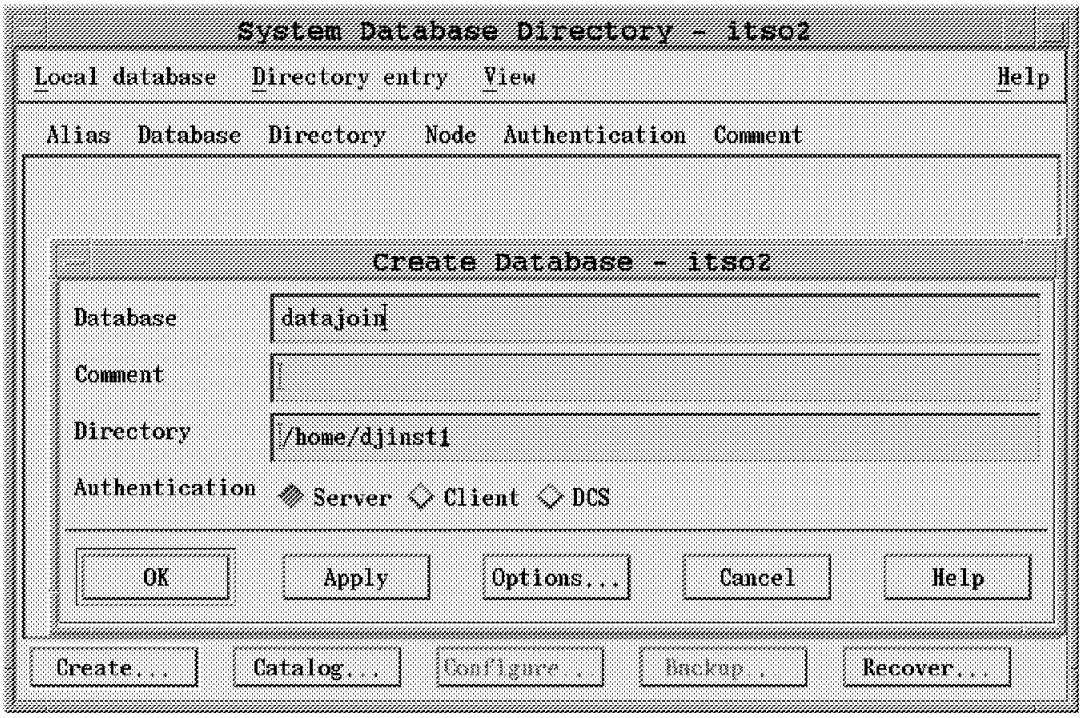


Figure 22. DBA Utility Window (Create Database)

Fill in the database name (`datajoin` in the example).

After you click on the OK button, the system creates the database. Creation takes a few minutes.

When the database is created, you then see the window shown in Figure 23 on page 35.

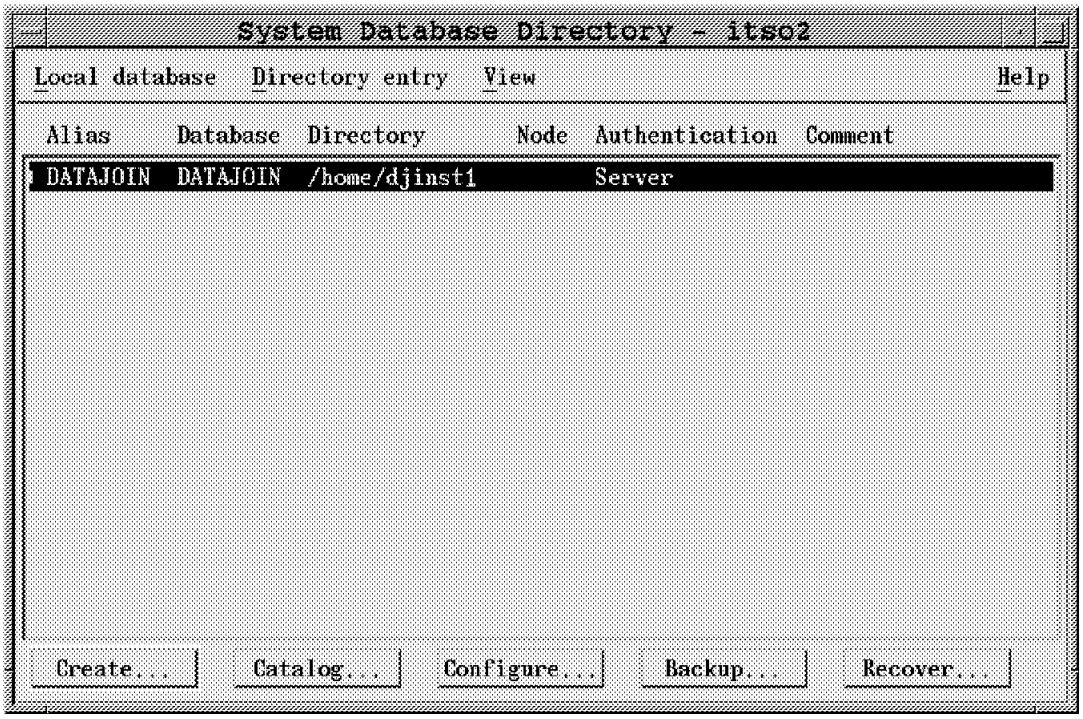


Figure 23. DBA Utility Window (List Database). Overview

At this point you can update the Communication Catalog Tables as described in more detail in Chapter 5, “Configuration of DataJoiner for Data Sources” on page 79.

3.3.2 Configuring DataJoiner for TCP/IP Clients

DataJoiner can be connected to the clients and data sources using different protocols. The clients can communicate to DataJoiner with TCP/IP or APPC. The data sources can communicate with DataJoiner using TCP/IP or SNA protocol.

You must have root authority to configure DataJoiner for using TCP/IP connections from remote clients.

The DataJoiner machine used for the examples has a host name, `itso1`, and the internet protocol (IP) address 192.200.111.88. We refer to this machine as `itso1`. The steps to configure TCP/IP are as follows:

3.3.2.1 DataJoiner TCP/IP Client Correlation

Appendix J, “TCP/IP Correlations and Worksheets” on page 243 is an example of a DataJoiner client correlation sheet from our installation. Figure 122 on page 244 is a DataJoiner client correlation worksheet that you can use for your installation.

3.3.2.2 Creation of a Service Name

The steps are these:

1. Select a unique pair of port numbers that are not defined on either machine in your configuration. The port numbers must be consecutive and must be greater than 1024.

2. Check the /etc/ services or equivalent file on both the DataJoiner machine and the client machine. Choose an unused service name for each of the two ports. It is preferable to use names that are unique on both sites (the names need not be the same on the client and the data source).

3. Login as user root.

   Change the /etc/ services file.

   For example, if ports 2455 and 2456 are unused in your existing configuration, then add lines to your /etc/services file as shown in Figure 24

```
#
# Ports for DataJoiner
#
djxtcpip        2455/tcp                        # DataJoiner TCP/IP
djxinter        2456/tcp                        # DataJoiner interrupt
```

Figure 24. Example of /etc/services File

   Similar changes must be made for each client that will be connected using TCP/IP to DataJoiner (see 4.2, “Configuration of TCP/IP Clients” on page 50).

3.3.2.3 Synchronization of the /etc/services file

To synchronize the /etc/services file, the inetd daemon must reread it. Use the following procedure:

1. Login as user root.
2. Enter `inetimp`
3. Enter `refresh -s inetd`

3.3.2.4 Update of the Database Manager Configuration

To register the connection on the DataJoiner server workstation, you must update the database manager configuration file. Use the following procedure:

1. Login as the instance owner.
2. Update the database manager configuration file by adding the TCP/IP service name. Use the following command:

   `db2 update database manager configuration using svcename <portname>`

   where <portname> is the service name you previously defined in the /etc/services file.

   Here is an example:

   `db2 update database manager configuration using svcename djxtcpip`

3.3.2.5 Stopping and Starting DataJoiner

You must be logged in as the instance owner to carry out the following steps:

1. If DataJoiner is already started, then enter

   `db2stop`

2. Now enter

   `db2start`

   to complete the update of the configuration of the server.

3. Check that your changes were completed successfully with the following command:

   ```
   db2 get database manager configuration
   ```

   Check that the Service name (SVCENAME) field contains the name you specified when you updated the database manager configuration. Figure 25 shows an example.

```
         Database Manager Configuration

Database manager configuration release level  = 0x0500
Node type                                     = Server with remote clients

Service name                                 (SVCENAME) = djxtcpip
Transaction program name                       (TPNAME) =

Max requester I/O block size (bytes)         (RQRIOBLK) = 4096
Max server I/O block size (bytes)            (SVRIOBLK) = 4096
Communication heap size (4KB)               (COMHEAPSZ) = 128
Remote services heap size (4KB)              (RSHEAPSZ) = 128
Sort heap threshold (4KB)                   (SHEAPTHRES) = 4096
Application support layer heap size (4KB)    (ASLHEAPSZ) = 100

Max no. of existing agents                   (MAXAGENTS) = 200
Max no. of concurrent agents                (MAXCAGENTS) = MAXAGENTS
Max no. of concurrently active databases         (NUMDB) = 8

Application cleanup interval (ms)           (CUINTERVAL) = 5000
Keep DARI process                             (KEEPDARI) = YES
Max. no. of DARI processes                     (MAXDARI) = MAXAGENTS
Priority of agents                            (AGENTPRI) = SYSTEM
Database monitor SQL statement size (bytes) (SQLSTMTSZ) = 256
Index re-creation time                         (INDEXREC) = RESTART
Default database path                         (DFTDBPATH) = /home/djinst1

Backup buffer default size (4KB)              (BACKBUFSZ) = 1024
Restore buffer default size (4KB)             (RESTBUFSZ) = 1024
```

Figure 25. Output of DB2 Get Database Manager Configuration

3.3.3 Configuring DataJoiner for APPC Clients

3.3.3.1 Introduction

In this section on configuring DataJoiner for APPC clients, we assume that SNA Server/6000 has already been installed on the DataJoiner machine and that the following have been defined in SNA Server/6000:

- Initial Node Setup
- Data-Link Control Profile
- Logical Unit (LU) 6.2 Mode Profile.

These steps are described in 5.1, “Connecting DataJoiner to DRDA-Attached Data Sources” on page 80.

To enable access by APPC clients, you also need to do the following:

1. Define a transaction program name profile to SNA Server/6000

2. Consider APPC security
3. Update the database manager configuration with the transaction program name.

3.3.3.2 Define Transaction Program Name Profile to SNA Server/6000

The LU 6.2 transaction program name (TPN) profile defines a combination of AIX and SNA characteristics of a transaction program on the system where DataJoiner is installed. Depending upon the types of clients that access DataJoiner, you may need to define up to four LU 6.2 TPN profiles on the machine where DataJoiner resides (see Table 2 for TPN values).

To access a database using APPC, there must be a process to handle database connections and another to handle database interrupts. Earlier versions divided this function into two separate transaction programs. More recently, both functions have been included in one transaction program.

DB2/2 V1 and OS/2 Extended Services V1.3 both use separate transaction programs, each with hard-coded hexadecimal TPNs. Therefore, if DataJoiner is to support these clients, the two separate hexadecimal transaction program names must be defined at the DataJoiner level.

Client Application Enabler/x V1 operates slightly differently. It still uses two separate transaction names. The name of the interrupt program is fixed at DB2INTERRUPT. You can give the connection program any name you want, as long as the name is the same for both the client and DataJoiner.

DB2/2 V2 and CAE/x V2 use a transaction program which combines the connection function and the interrupt function in the same program. As with CAE/x V1, you may choose any name that you want for this transaction program.

Table 2. TPN Values

Parameter	**CAE/x & DB2/2 V2**	**CAE/x V1**	**DB2/2 V1.x & OS/2 ES V1.3**	
Profile Name 1	zzservertp	zzserverint	zzserveros2tp	zzserveros2int
TPN 2	OS2TP	DB2INTERRUPT	07F6C4C2	07F6E2D5
TPN in hex?	No	No	Yes	Yes
Conversation type	basic	basic	basic	basic
Path 3	/db2acntp	/db2aittp	/db2acntp	/db2cnsm
Multiple instances	yes	yes	yes	yes
User ID 4	1000	1000	1000	1000

NOTES

1 The profile name used is arbitrary. It is a tag used by SNA Server/6000 to group the information together as a unit. It is not to be confused with the TPN, which is the actual value that will be passed from the client. However, there is nothing to prevent you from using the same value for Profile Name and TPN.

2 CAE/x clients and DB2/2 V2 clients define their APPC connection to servers by using the CPI-C Side Information facility of Communication Manager/2.

Therefore you have the flexibility of specifying any value that you like for the TPN, as long as the value chosen matches the TPN at the server. However, the TPN for the interrupt program must be DB2INTERRUPT.

DB2/2 V1.x and OS/2 Extended Services V1.3 clients do not have the same flexibility. Both the TPN for the DB2 program and the TPN for the interrupt program are hard-coded hexadecimal values. These hexadecimal values must be used when defining the transaction program name profiles at SNA Server/6000 and Communication Manager/2.

3 The value represented in Table 2 on page 38 is the name of the executable program. In fact, the Full PATH to TPN executable field must be `INST/sqllib/bin/xxxxxxxx` where INST is the home directory for the instance owner and xxxxxxxx is the name of the path found in Table 2 on page 38.

4 The User ID field must be set to the instance owner's user ID number (not the user name). This value can be obtained by issuing the AIX "id" command while logged on as the instance owner.

3.3.3.3 DataJoiner APPC Client Correlation

Appendix K, "APPC (SNA) Correlations and Worksheets" on page 247 shows an example of the DataJoiner client correlation sheet from our installation. A DataJoiner client correlation worksheet, which you can use for your installation, is shown in Figure 126 on page 248.

3.3.3.4 Sample TPN Profile Definitions

The TPN definitions in Figure 26 on page 40, Figure 27 on page 41, Figure 28 on page 42, and Figure 29 on page 43 show the details of each of the profiles referred to in Table 2 on page 38.

```
                        Change/Show LU 6.2 TPN Profile

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

                                                      [Entry Fields]
  Current profile name                                zzservertp
  New profile name                                   []
  Transaction program name (TPN)                     [OS2TP]
  Transaction program name (TPN) is in hexadecimal?   no                      +
  PIP data?                                           no                      +
     If yes, Subfields (0-99)                        [0]                       #
  Conversation type                                   basic                   +
  Sync level                                          none/confirm            +
  Resource security level                             none                    +
     If access, Resource Security Access List Prof.  []
  Full path to TP executable         [/home/djinst1/sqllib/bin/db2acntp]
  Multiple instances supported?                       yes                     +
  User ID                                            [1000]                    #
  Server synonym name                                []
  Restart action                                      once                    +
  Communication type                                  signals                 +
     If IPC, Communication IPC queue key             [0]                       #
  Standard input file/device                         [/dev/console]
  Standard output file/device                        [/dev/console]
  Standard error file/device                         [/dev/console]

  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. TPN Profile Definition for OS2TP

The TPN profile in Figure 26 is used to handle database connections and database interrupts from DB2 Client Application Enabler Version 2 clients, and database connections from DB2 Client Application Enabler Version 1 clients.

Database interrupts from DB2 Client Application Enabler Version 1 clients are handled using the TPN profile in Figure 27 on page 41.

```
                        Change/Show LU 6.2 TPN Profile

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

                                                      [Entry Fields]
  Current profile name                                zzserverint
  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]                       #
  Conversation type                                   basic                   +
  Sync level                                          none/confirm            +
  Resource security level                             none                    +
     If access, Resource Security Access List Prof.  []
  Full path to TP executable         [/home/djinst1/sqllib/bin/db2aittp]
  Multiple instances supported?                       yes                     +
  User ID                                            [1000]                    #
  Server synonym name                                []
  Restart action                                      once                    +
  Communication type                                  signals                 +
     If IPC, Communication IPC queue key             [0]                       #
  Standard input file/device                         [/dev/console]
  Standard output file/device                        [/dev/console]
  Standard error file/device                         [/dev/console]

  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. TPN Profile Definition for DB2INTERRUPT

The TPN profile in Figure 28 on page 42 is used to handle database connections from DB2/2 Version 1 clients.

```
                        Change/Show LU 6.2 TPN Profile

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

                                                      [Entry Fields]
  Current profile name                                zzserveros2tp
  New profile name                                   []
  Transaction program name (TPN)                     [07F6C4C2]
  Transaction program name (TPN) is in hexadecimal?   yes                    +
  PIP data?                                           no                     +
     If yes, Subfields (0-99)                        [0]                      #
  Conversation type                                   basic                  +
  Sync level                                          none/confirm           +
  Resource security level                             none                   +
     If access, Resource Security Access List Prof.  []
  Full path to TP executable         [/home/djinst1/sqllib/bin/db2acntp]
  Multiple instances supported?                       yes                    +
  User ID                                            [1000]                   #
  Server synonym name                                []
  Restart action                                      once                   +
  Communication type                                  signals                +
     If IPC, Communication IPC queue key             [0]                      #
  Standard input file/device                         [/dev/console]
  Standard output file/device                        [/dev/console]
  Standard error file/device                         [/dev/console]

  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. TPN Profile Definition for X'07'6DB

The TPN profile in Figure 29 on page 43 is used to handle database interrupts from DB2/2 Version 1 clients.

```
                        Change/Show LU 6.2 TPN Profile

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

                                                      [Entry Fields]
  Current profile name                               zzserveros2int
  New profile name                                   []
  Transaction program name (TPN)                     [07F6E2D5]
  Transaction program name (TPN) is in hexadecimal?   yes                    +
  PIP data?                                           no                     +
     If yes, Subfields (0-99)                        [0]                      #
  Conversation type                                   basic                  +
  Sync level                                          none/confirm           +
  Resource security level                             none                   +
     If access, Resource Security Access List Prof.  []
  Full path to TP executable         [/home/djinst1/sqllib/bin/db2cnsm]
  Multiple instances supported?                       yes                    +
  User ID                                            [1000]                   #
  Server synonym name                                []
  Restart action                                      once                   +
  Communication type                                  signals                +
     If IPC, Communication IPC queue key             [0]                      #
  Standard input file/device                         [/dev/console]
  Standard output file/device                        [/dev/console]
  Standard error file/device                         [/dev/console]

  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. TPN Profile Definition for X'07'6SN

The TPN profile in Figure 29 is used to handle database interrupts from DB2/2 Version 1 clients.

3.3.3.5 APPC Security Considerations

In addition to the above, SNA Server/6000 should be configured to accept the type of security requested by the client. Security considerations are discussed in detail in Chapter 6, “Security” on page 145.

3.3.3.6 Update of the Database Manager Configuration with the TPN

The transaction program name in the *tpname* parameter in the database manager configuration file must be the same as the transaction program name configured in the TPN profile of SNA Server/6000, which handles database connections from the clients. Using the information in Table 2 on page 38 you can see that the TPN in our setup is *OS2TP*.

To update, using the login ID of the DataJoiner instance owner, take the following steps:

1. Use the 'db2adm' utility to update the *tpname* value of the Database Manager Configuration File. This value must be the same as the TPN that was defined in the TPN Profile for remote clients connect to DataJoiner,

   or

Use the CLP as follows: `update database manager configuration using tpname OS2TP`

2. Stop all connections to the instance by either issuing

   ```
   db2 force applications all
   ```

   or

   ```
   db2stop
   ```

3. Restart DB2 by issuing the `db2start` command.

3.3.4 Removing a DataJoiner Database

To remove a database you may have created for testing, use the `drop database` command. This command deletes all data files, all user files, and the database definition for the database. DataJoiner will then uncatalog the entry from the database manager configuration file and local database directory. You can drop only local databases. All users must be disconnected from a database before it is dropped.

The `drop database` command can be run from the command line, from the CLP prompt, or by using the DBA tool.

3.3.4.1 Using the Command line

The format of the command is:

```
db2 drop database <database name>
```

where `<database name>` is the name of the database you want to drop.

Here is an example:

```
db2 drop database sample
```

3.3.4.2 Using the DBA Tool

To use the DBA Tool you need AIXwindows.

To start this tool, from an AIXwindows terminal, type:

```
db2adm &
```

A window is then displayed as shown in Figure 30.

Figure 30. DBA Utility Window (Main)

To select the database you want to remove, click on the **open** button. You then see a window such as that shown in Figure 31 on page 45.

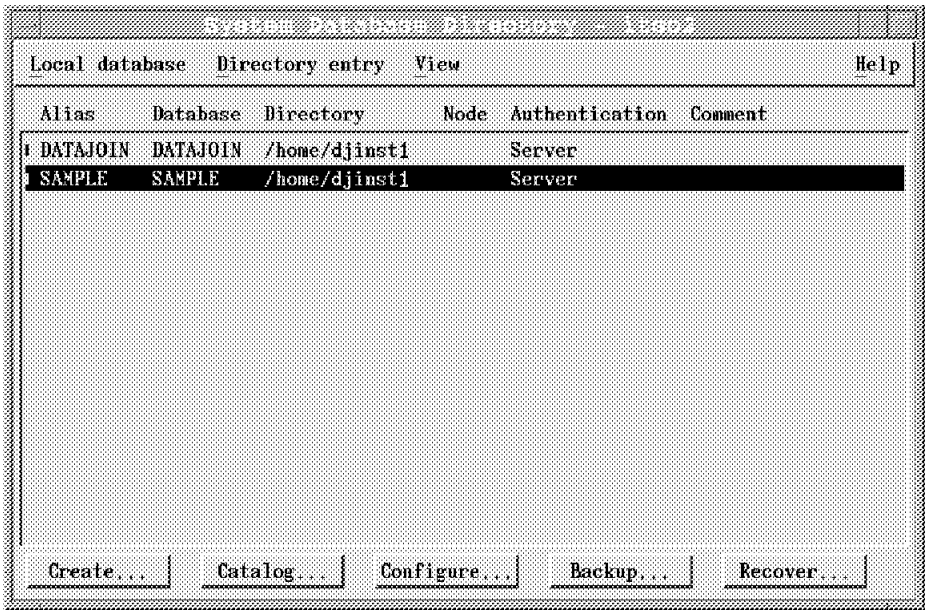


Figure 31. DBA Utility Window (Select Database)

Select the database you want to delete and click on **Local database**. The result is a window such as that shown in Figure 32.

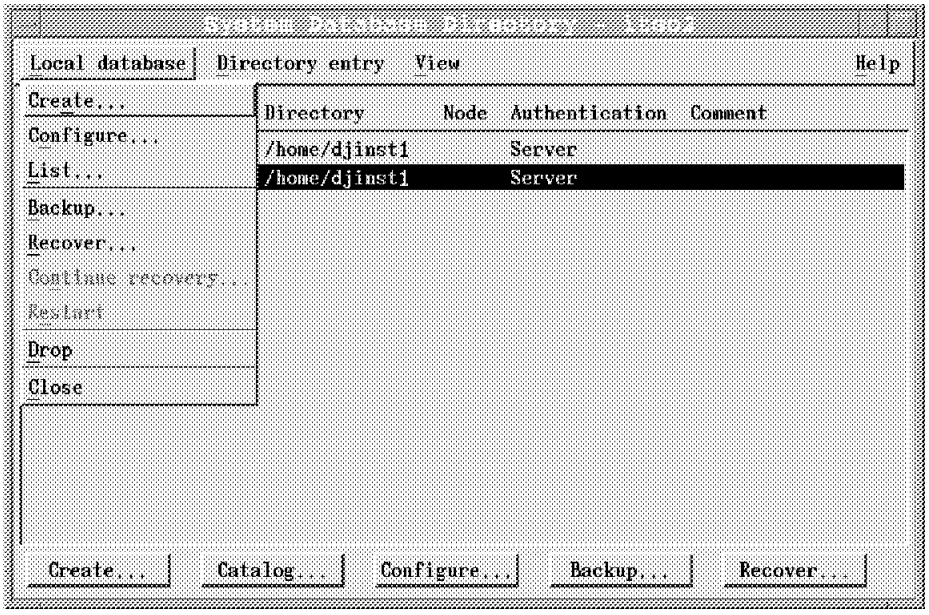


Figure 32. DBA Utility Window (Select Popup)

Select **Drop**. A confirmation popup window appears as shown in Figure 33 on page 46.

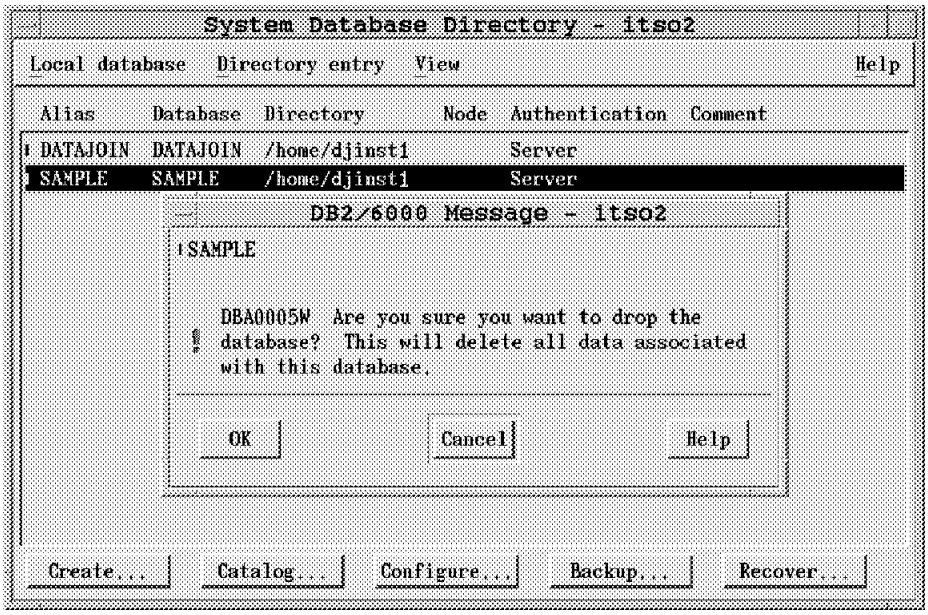


Figure 33. DBA Utility Window (Confirmation Popup)

Click on the **OK** button to drop the database. When the database is dropped, you see the window shown in Figure 31 on page 45 again, but without the database you had selected to be dropped.

Chapter 4. Configuration of DataJoiner Clients

This chapter provides information on and examples of configuration of various types of clients that can access data through DataJoiner.

4.1 Overview

Depending on the client software, you use different types of protocols for communication.

Figure 34 shows the clients involved in the configuration.

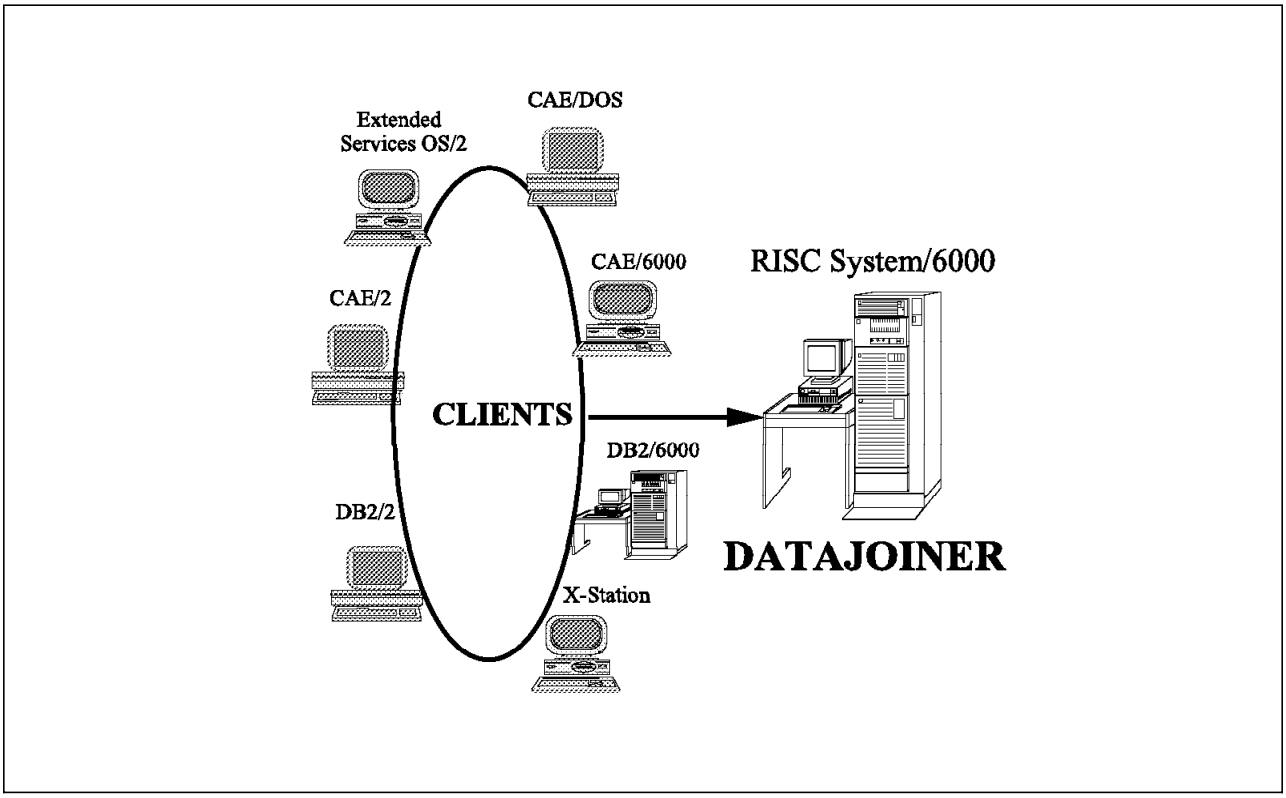


Figure 34. Clients Connected to DataJoiner

The client configurations differ depending on whether the client software is based on OS/2, DOS, AIX, or X-Station.

An OS/2 client can be connected using TCP/IP or APPC to DataJoiner. There are different software requirements for these connections. For example, APPC can be connected using DB2/2 or Client Application Enabler (CAE/2).

A DOS client can be connected solely by using TCP/IP.

When connecting an AIX system to DataJoiner, you can use TCP/IP or APPC with either CAE/6000 or DB2/6000. An X-Station can be used with an AIX client or with the DataJoiner system itself.

Figure 35 on page 48 shows the protocols and their relationships to the various clients.

© Copyright IBM Corp. 1995

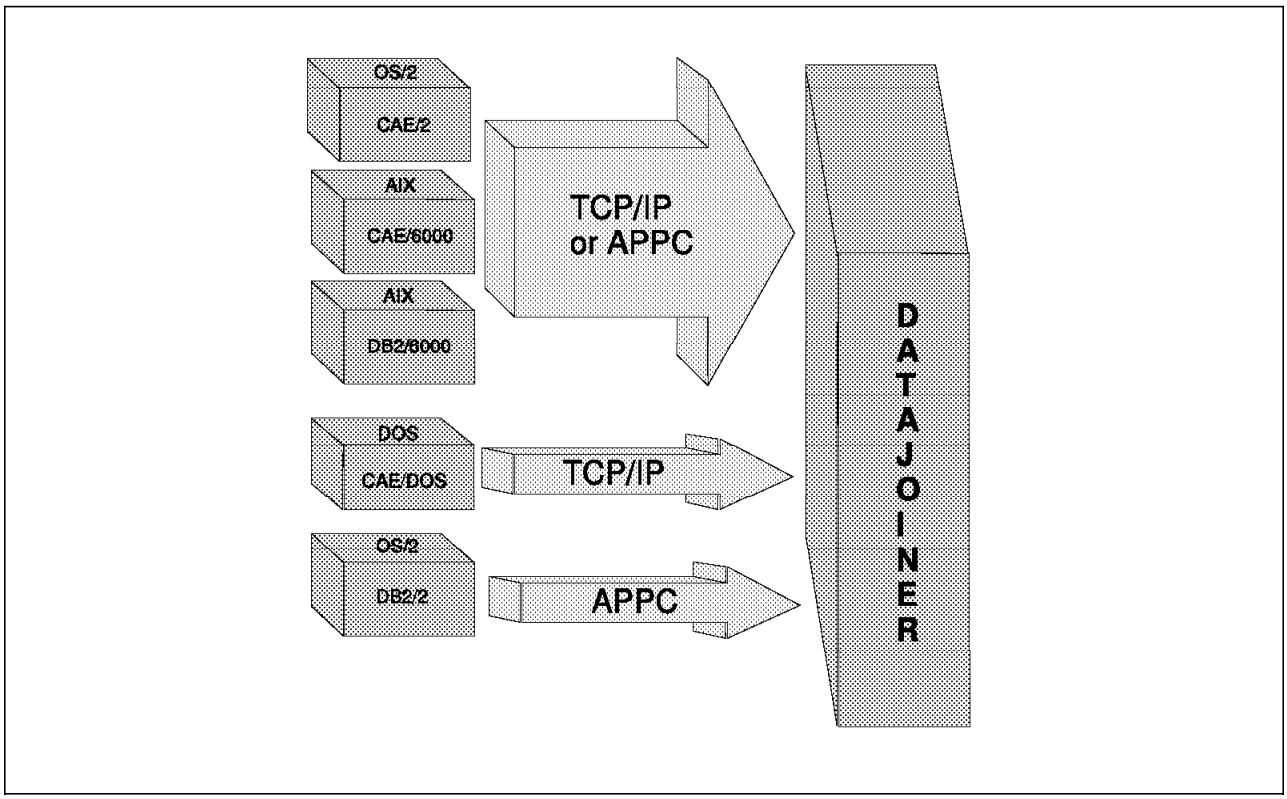


Figure 35. Protocols Used to Connect the Clients to DataJoiner

4.1.1 Client Support

This component allows a DB2 server to accept requests from remote clients using several communication protocols. DB2/2 includes *client support* as an integral part of the DBMS. Client support is not included with DB2/6000 but is a separate component.

Client support enables the workstation DB2 products to accept requests from remote clients.

DB2/2 Version 1 can only be used as an application requester (client), but DB2/2 Version 2 can be used as application requester or application server. DB2/2 Version 1 cannot use TCP/IP protocol for communication, but DB2/2 Version 2 can.

DataJoiner looks like a DB2/6000 database and therefore can be accessed using either APPC or TCP/IP protocol.

4.1.2 Client Application Enabler (CAE/2 or CAE/6000)

Client Application Enabler provides run-time support for DB2 database client applications. It allows access to remote DB2/2 or DB2/6000 server systems. Applications can be developed using embedded SQL, database manager application program interfaces (APIs) and DB2 call-level interface APIs. CAE also provides an Open Database Connectivity (ODBC) driver.

You should use the specific CAE product for your client environment—OS/2, DOS/Windows, or AIX. CAE/2 includes functions that allow you to connect to remote servers through APPC TCP/IP and NetBIOS. The DB2 Client Application

Enabler for DOS and Windows (CAE/DOS) includes functions that allow you to connect to remote servers (DB2/2 and DB2/6000) through APPC, TCP/IP, NetBIOS and IPX/SPX. The DB2 Client Application Enabler/6000 for AIX (CAE/6000) includes functions that allow you to connect to remote servers through APPC and TCP/IP.

4.1.3 DB2 Software Developer′s Kit (SDK)

SDK provides all the functions found in the DB2 Client Application Enabler (see 4.1.2, “Client Application Enabler (CAE/2 or CAE/6000)” on page 48) as well as a full application development environment for the client workstation. SDK also includes code examples, precompilers, and other development tools.

4.2 Configuration of TCP/IP Clients

This section provides information on and examples of the details of specifying connections from clients that use the TCP/IP communications protocol.

4.2.1 Configuration of OS/2 or DOS/Windows Clients

To use the TCP/IP protocol from an OS/2 or DOS/Windows client to DataJoiner you should install the corresponding TCP/IP product and install the corresponding CAE or SDK product.

To use DOS, OS/2, or WIN-OS/2 programs to access DataJoiner, you must install the TCP/IP Access Kit for DOS and WINDOWS and CAE/DOS.

4.2.1.1 TCP/IP Configuration

During or after installation, you should configure TCP/IP for each client. Each client must have a *unique* host name and IP address in your network. If you are using a router (gateway), you also need to specify the gateway IP address.

When you install TCP/IP, you can specify the host name, IP address and gateway IP address during installation of TCP/IP as shown in Figure 36 which illustrates TCP/IP in the OS/2 environment.

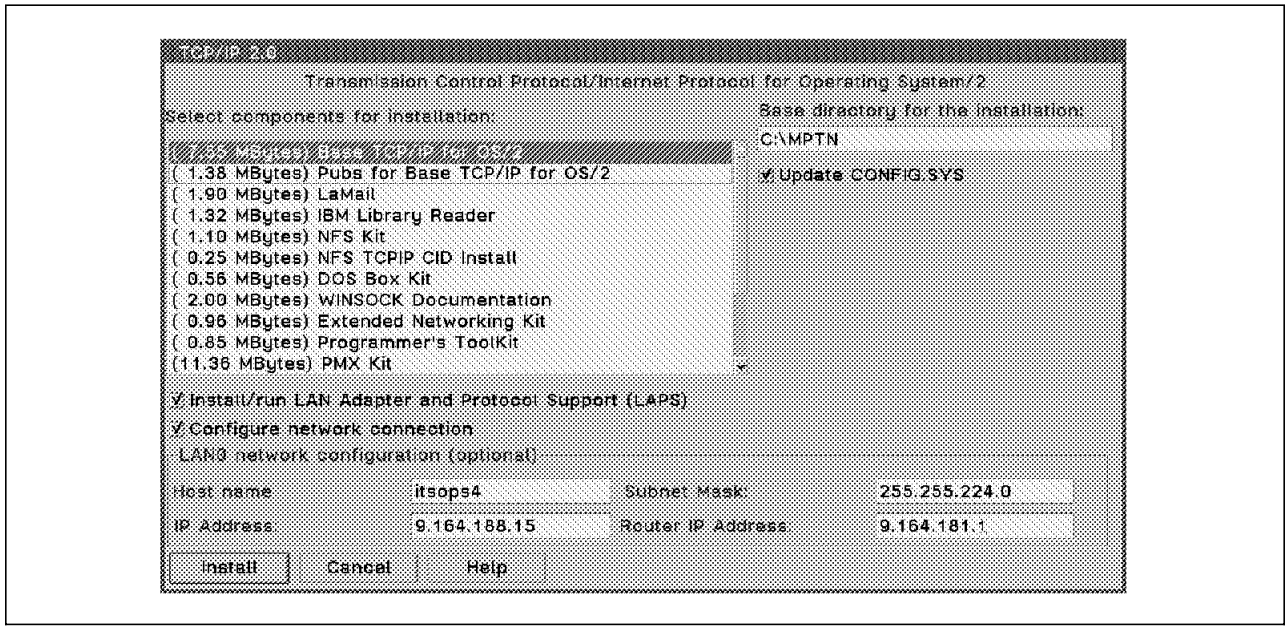


Figure 36. Initial Window for TCP/IP Installation

You can carry out the configuration after installing TCP/IP using the TCPIPCFG program or the **Configuration** icon.

Next, you should edit the TCPIP\ETC\services file and add the ports already defined on the RISC System/6000 (see 3.3.2, “Configuring DataJoiner for TCP/IP Clients” on page 35) as in Figure 37 on page 51.

```
    .
    .
    .
#  DataJoiner in Boeblingen ports for connections   #
djxtcpip       2455/tcp     # port to access DataJoiner via tcpip
djxinter       2455/tcp     # interrupt port
```

Figure 37. Example of Client Port Definition

At this point the client can connect to the AIX system. To run DB2 commands on other databases cataloged in DataJoiner, you need to update the system database by:

- Cataloging a TCP/IP node
- Cataloging a database
- Running a bind command if necessary.

4.2.1.2 Update the System Catalog

You should run all of these commands from a DB2 command prompt or, if you use an OS/2 command prompt, prefix the command with db2.

The following example shows the cataloging of a TCP/IP node for our DataJoiner machine which has the host name **itso1**:

```
catalog tcpip node nodedj remote itso1 server djxtcpip with "dj-node in germany"
```

where

nodedj	Node entry name for use in the system database directory
itso1	Host name for the system on which DataJoiner is installed (in Germany in our example)
djxtcpip	service name (port name) defined in the services file (see 4.2.1.1, "TCP/IP Configuration" on page 50).

You should next catalog the DataJoiner database (DataJoiner is in fact a DB2/6000 database management system with special functions). Here is an example:

```
catalog database datajoin as boedj at node nodedj authentication xxx
          with "DataJoiner database"
```

where

datajoin	Name of a database under a DataJoiner instance.
boedj	Alternate name (alias) for the database that is being cataloged
nodedj	Specifies the node where DataJoiner server resides. This must match the name you specified in the node directory.
xxx	Is an authentication parameter which can be specified as server, dcs, or client.

In our case, we create the database with authentication set to server. For more details on authentication, see 6.2.5, “Authentication Parameters” on page 148.

Before using DataJoiner (known in our client as **boedj** you must connect to it.

To connect successfully to DataJoiner, you need a user ID and password in the AIX system that was defined to access DataJoiner. Otherwise, you get an SQL error (-1403). Here is an example of a CONNECT statement:

```
connect to boedj user userid using  password
```

where

boedj	Name of a database under a DataJoiner instance.
userid	Valid *user ID* on the system (itso1 in our example) which can access DataJoiner.
password	Valid *password* defined (in upper case) on the system (itso1 in the example). See Chapter 6, “Security” on page 145.

After the connection is made, you must under certain conditions run a bind command. Each time you create a new database on your server, you must create packages for the database utilities. You can create these packages by running the bind command once, from each type of client: CAE/2 Version 1 clients, CAE/2 Version 2 clients, CAE/DOS Version 1 clients, or DB2/2 Version 1 clients. Here is an example of the bind command:

```
bind \sqllib\bnd\@db2ubind.lst blocking all
```

While CAE/DOS Version 1.2 can be installed in the same partition as CAE/2 Version 1.2, DB2/2 Version 1 and CAE/2 Version 1.2 cannot reside in the same partition.

4.2.1.3 A Specific Example: Connecting Visualizer Query for Windows to DataJoiner

To connect Visualizer Query For Windows (VQW) to DataJoiner, you must first define the connection between CAE/DOS and DataJoiner as described in 4.2.1, “Configuration of OS/2 or DOS/Windows Clients” on page 50. Figure 38 on page 53 shows the environment that we implemented to connect VQW to DataJoiner.

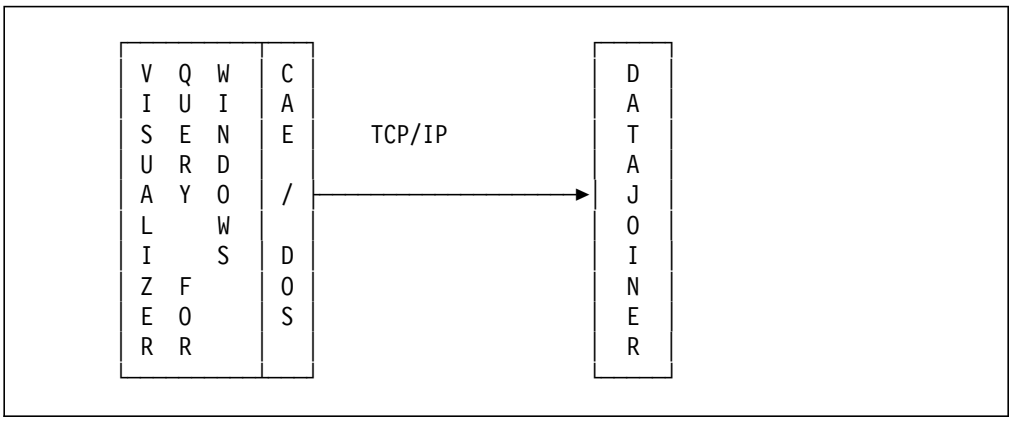


Figure 38. Visualizer Query for Windows Connectivity Scenario

The steps to connect VQW to DataJoiner are:

1. Install CAE/DOS.
2. Install VQW.
3. Define the ODBC data source.
4. Define the database schema.

Install CAE/DOS: When you install CAE/DOS, select the following options from the installation screens:

- TCP/IP protocol support
- DOS and Windows environment
- Automatic update of the AUTOEXEC.BAT file.

Install Visualizer Query for Windows: When you install VQW, select the following components:

- Visualizer Query for Windows Query
- Visualizer Query for Windows Administrator.

Define the ODBC Data Source: Before defining the ODBC data source, set the DB2CODEPAGE environment variable in the AUTOEXEC.BAT file to match the code page of the database you are going to use. For example, you can set this variable as follows:

```
SET DB2CODEPAGE=850
```

To make the databases on the DataJoiner server known to the Windows client, issue commands from the CAE command line. In the following example, a database called SAMPLE is cataloged:

1. Open a DOS Window and enter:

   ```
   db2
   ```

2. Give the client machine a name. In the following example, the client machine is named client1.

   ```
   update database manager configuration using nname client1
   ```

3. Catalog the node in which DataJoiner resides. Use the host name for the DataJoiner server that you defined in the HOSTS file of the client, and the service name that relates to the service number on DataJoiner that is

defined to receive requests from TCP/IP clients. In this example, the host name is severn and the service name is djsrv.

```
catalog tcpip node dj remote severn server djsrv
```

4. Catalog the database called SAMPLE, as in this example:

```
catalog database sample as sample at node dj
```

5. Bind the database called SAMPLE, as in this example:

```
connect to sample user djinst1 using password
bind c:\caedos\bnd\@db2ubind.lst blocking all grant public
quit
```

Return to the Windows environment and register the ODBC data source by executing the command file called DB2ODBC.EXE that is in the directory where you installed CAE.

To make the new database accessible through ODBC, do the following:

1. From the Program Manager, open the Main Group.
2. Open Control Panel.
3. Open ODBC.
4. Click on the Add button on the Data Sources window.
5. Select Client Application Enabler/DOS from Add Data Sources.
6. Select the database (called SAMPLE in our scenario) from the drop-down list of the CAE/DOS ODBC setup window.
7. Enter a description and click on OK.

Define the Database Schema:

1. Go to the VQW Administrator and open a new schema.
2. On the ODBC Data Sources window, select the SAMPLE database.
3. On the Schema window, select the Add Table icon.
4. Enter the qualifier of the tables on the DataJoiner data source that you want to access.
5. Click on the search button of the drop-down list for the names of the tables for that qualifier.
6. Select the table you want to access, define a short and a long description, and click on OK.
7. Put the mouse pointer over the new table icon and click the right-hand button. Select Edit Columns Details from the pop-up menu.
8. On the Columns Details window, click the icon to retrieve the column definitions from the data source. A list of all the columns pertaining to that table should appear. You can then select the columns from that list.

4.2.2 Configuration of AIX Clients

In the ITSO implementation of AIX clients, we have an AIX client system with CAE/6000 and another client with DB2/6000, both connected by TCP/IP to DataJoiner.

As Datajoiner works with the same clients as a DB2/6000 Version 1 database, DataJoiner instances *can* coexist on the same workstation as DB2/6000 Version 1 or Version 2 instances.

To connect a client to DataJoiner, you need only to configure TCP/IP and update the system database.

4.2.2.1 TCP/IP Configuration

To configure TCP/IP, you must have root authority on the AIX client.

Follow these steps to configure TCP/IP on your AIX client:

1. Edit the /etc/services file, and add the same two port numbers used on the DataJoiner server (see 3.3.2, “Configuring DataJoiner for TCP/IP Clients” on page 35).
2. Synchronize the /etc/services file and the `inetd` daemon by executing the following commands from an AIX command line:

   ```
   inetimp
   refresh -s inetd
   ```

3. Optionally, update a domain name server with the TCP/IP hostname of your client.

4.2.2.2 Update the System Catalog

You must do the following steps with either system-administrator authority or instance-owner authority for the client.

The commands can be entered from a db2 command line, or from an AIXwindows session by prefixing each command with db2.

1. Catalog a local node pointing to the remote database (DataJoiner). Here is an example:

   ```
   catalog tcpip node djnode remote itso1 server djxtcpip
   ```

2. Catalog the remote database to the local database directory. Here is an example:

   ```
   catalog database datajoin as boedjt at node djnode authentication xxxx
   ```

For a description of the parameters, see 4.2.1.2, “Update the System Catalog” on page 51. For a description of authentication, see Chapter 6, “Security” on page 145.

Authorized AIX users on the client machine can now connect to DataJoiner.

4.3 Configuration of APPC Clients

In the ITSO environment, we use VTAM and therefore must specify definitions in VTAM to be able to connect the clients to the DataJoiner system (itso1 in our example).

For each client in VTAM, you must define

- A physical unit (PU) name for the client (Control Point)
- A logical unit (LU) name for the client
- A PU name for itso1 (the DataJoiner machine)

- An LU name for itso1.

If you are using APPC connections without VTAM, you have a peer-to-peer connection and you can choose your own unique names for:

- A Network ID
- A PU name for the client
- An LU name for the client
- A PU name for itso1 (the DataJoiner machine)
- An LU name for itso1.

To configure an APPC client, you need to use the unique LAN adapter addresses. Consult your LAN administrator to obtain these addresses.

4.3.1 Configuration for OS/2 Clients

OS/2 clients can have CAE/2, SDK/2, or DB2/2 installed.

Before beginning the configuration procedure, you need

- Communication Manager installed
- CAE/2, SDK/2, or DB2/2 installed (it is preferable to install this after installing CM/2)
- A LAN address for the client
- A LAN address for the DataJoiner system (itso1 in our example)
- A PU name for the client
- An LU name for the client
- The partner LU name of the DataJoiner system
- The transaction program name (for CAE/2 and DB2/2 Version 2 only).

To communicate with DataJoiner using APPC, you must

- Configure the Communication Manager.
- Update the system database.

The configuration steps to be done for a client with CAE/2 and DB2/2 are the same, except that for a client with DB2/2 Version 1, you do not configure the CPI Communications (as in Figure 59 on page 67). There is no need to do that because the transaction program name is hard coded.

The communication setup uses a TPN. For DB2/2 Version 1, the TPN is X'07'6DB and cannot be changed.

DataJoiner uses a transaction program with the name /home/djinst1/sqllib/bin/db2acntp. For this program, in our example we define two aliases, one for DB2/2 Version 1 and one for CAE/2 and DB2/2 Version 2 (called OS2TP).

For DB2/2 Version 2, you can use CPIC node definitions, so you can specify a name for the transaction program being used on DataJoiner (db2acntp or an alias defined in AIX to this program) similar to CAE/2. See Figure 59 on page 67.

4.3.1.1 Configure Communication Manager

The next pages will guide us step by step to configure Communication Manager/2 (CM/2) on an OS/2 machine for an APPC connection to DataJoiner, using the ITSO configuration as an example. See Figure 39 for the initial CM/2 window.

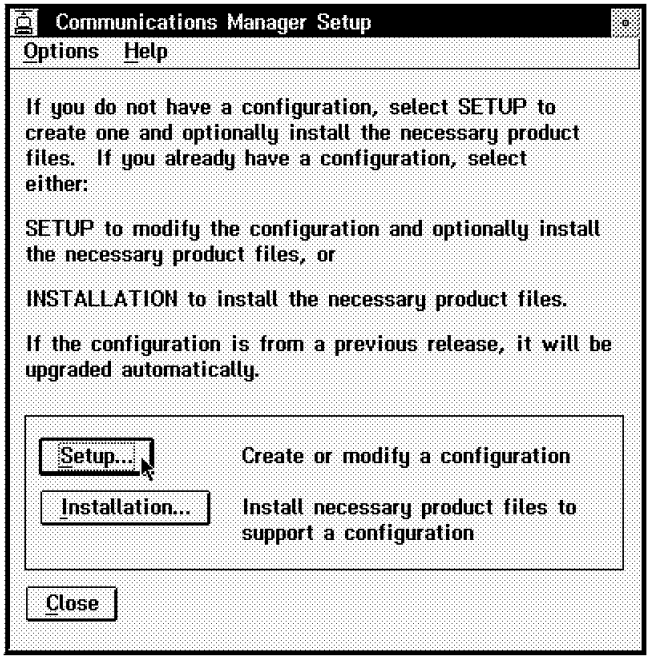


Figure 39. Communication Manager Setup

After selecting **Setup**, you see the window for opening a configuration as shown in Figure 40 on page 58. Note that the windows in the following examples may differ slightly from those on your system because of differences in the versions of CM/2 that are used.

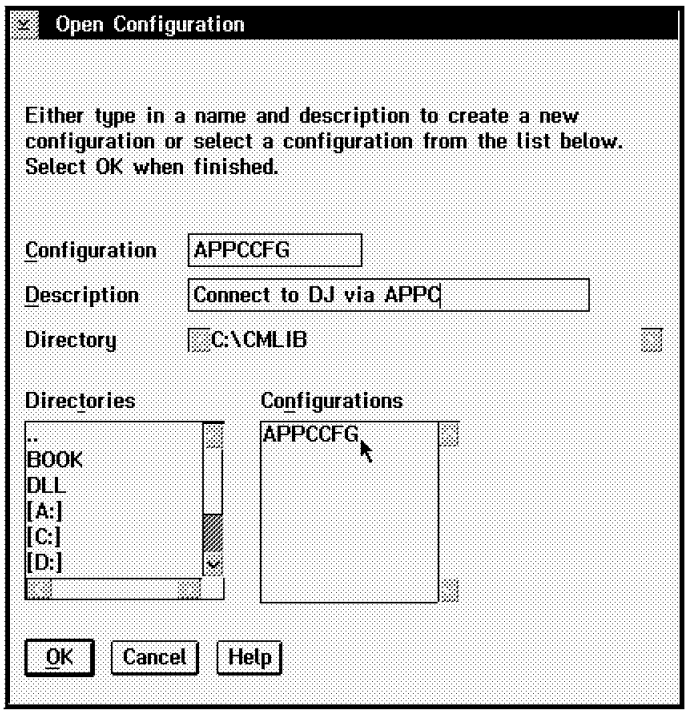


Figure 40. Create an APPC Configuration

You can use an existing configuration or create a new one (see Figure 40, Figure 41, and Figure 42 on page 59).

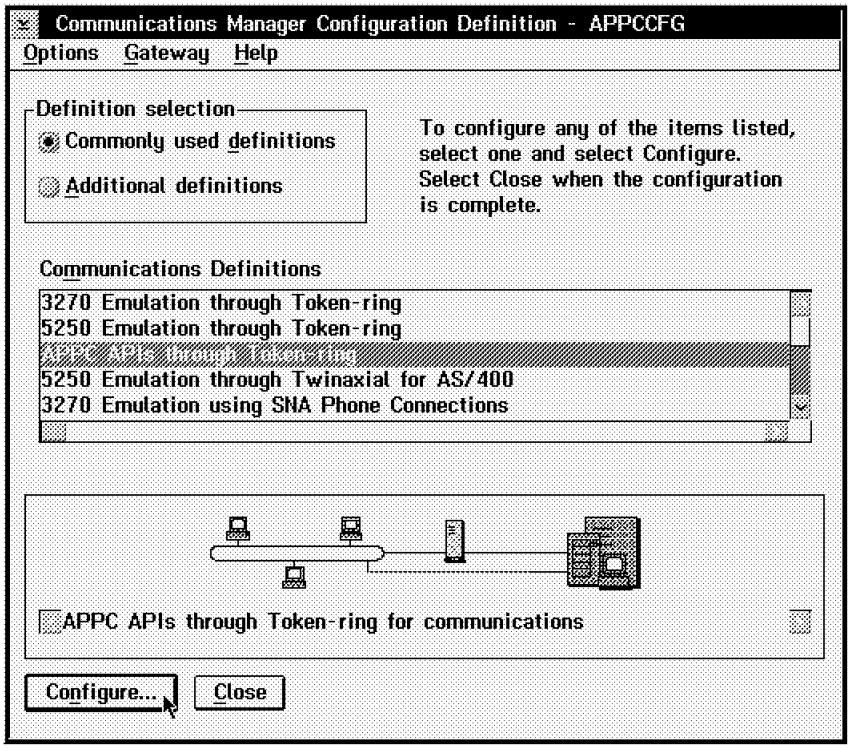


Figure 41. APPC Through Token Ring

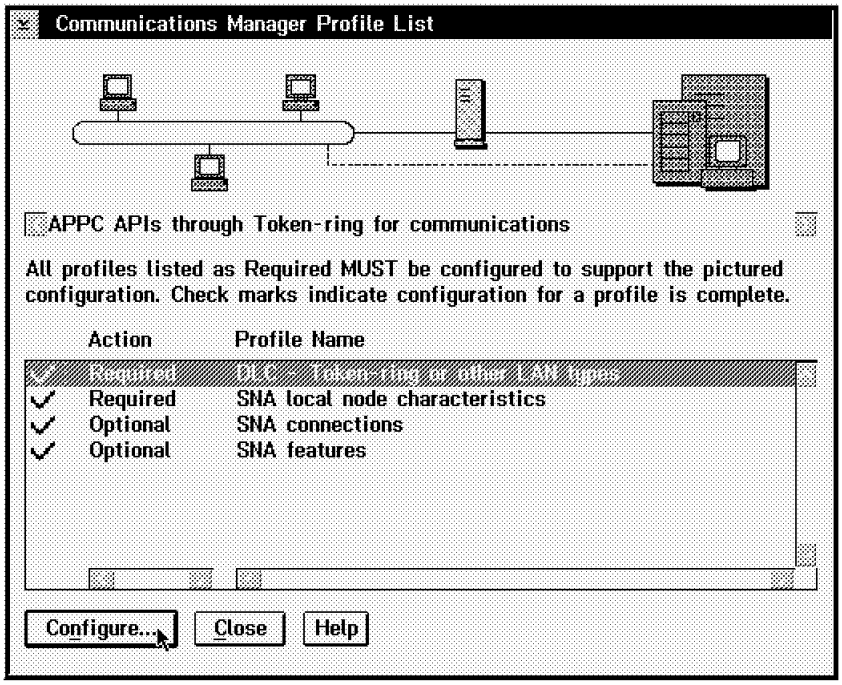


Figure 42. Settings for Token Ring

In our example, we used the default values for the LAN parameters. The adapter number can be set if the client has more than one adapter (see Figure 43).

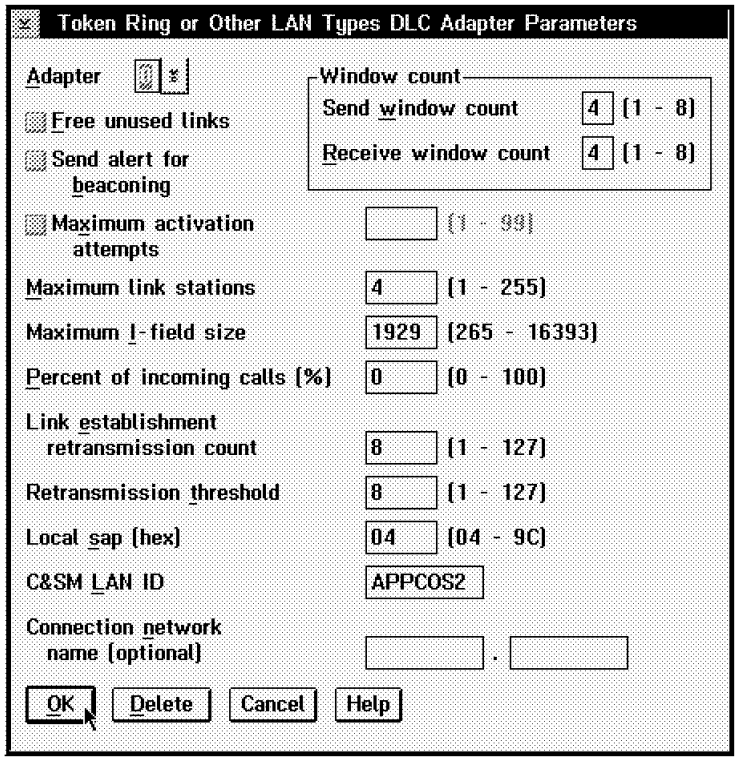


Figure 43. Token Ring Parameters

For the settings for SNA, see Figure 44 on page 60.

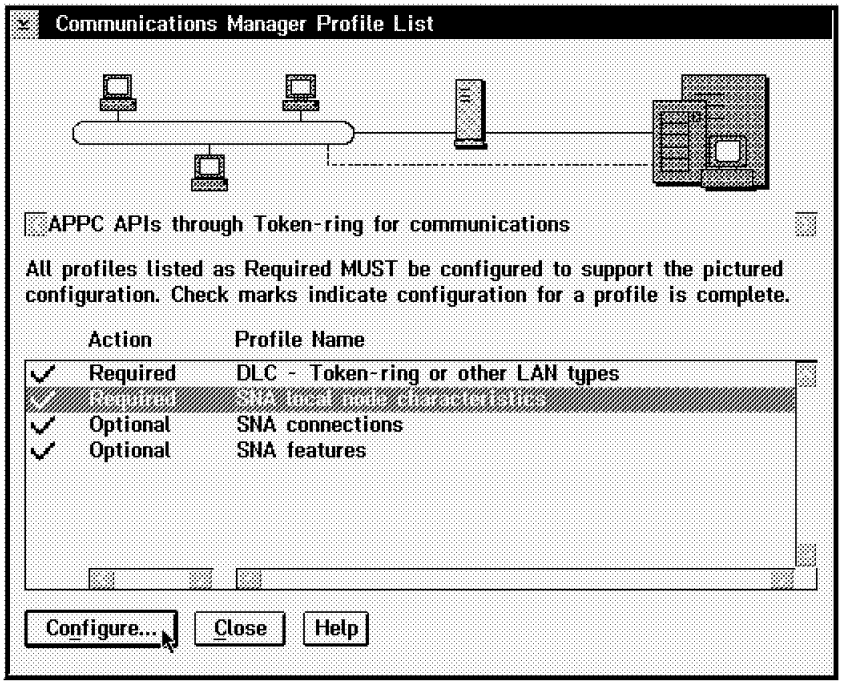


Figure 44. SNA Local Node Names

At this point, we configure our local node parameters, as shown in Figure 45.

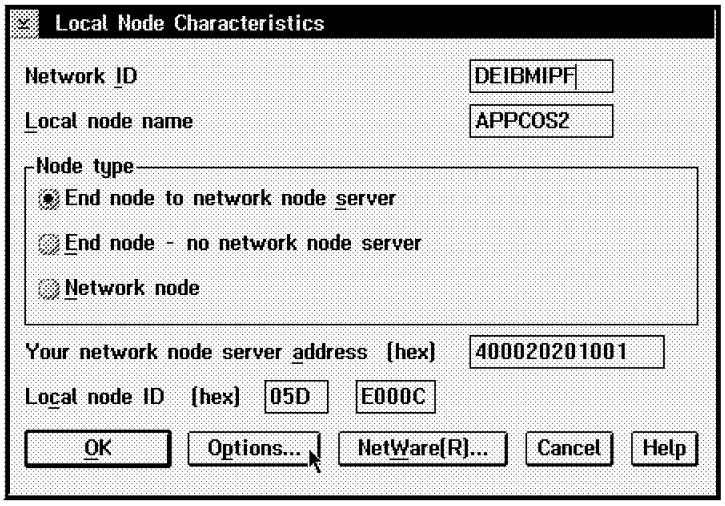


Figure 45. Local Node

The following example assumes an Advanced Peer-to-Peer Networking (APPN) connection. The **Node type** is End node to network node server. You must consider the additional parameters described below.

The **Network ID** is the identifier of the APPN network to which this machine is connected. The **Local node name** and **Local node ID** should be unique within each SNA network.

The **Local node name** specifies the name of the local control point (CP). If the local LUs are communicating to a subarea host as in our environment (for example, 3174 and 3745 controllers assigned to VM/ESA), they should be coordinated with the host (for example, VTAM PU definition).

When connecting to a host through a LAN, each workstation must have a unique **Local node ID**, and all the node IDs must be defined at the host. This corresponds to the IDNUM parameter in the VTAM PU definition. The first field (3 hexadecimal digits) is by default X'05D', which is the program ID for Communication Manager; the second field is the IDNUM in the VTAM PU definition. The **network node server address** in our case is the address from the 3174 controller in the network (LAN). For connectivity with 3745, choose a **to host** connection, as shown in Figure 48 on page 62.

The **Local node alias name** is a name that you can use instead of your local node name (it can be the same as or different from the local node name). Figure 46 is an example.

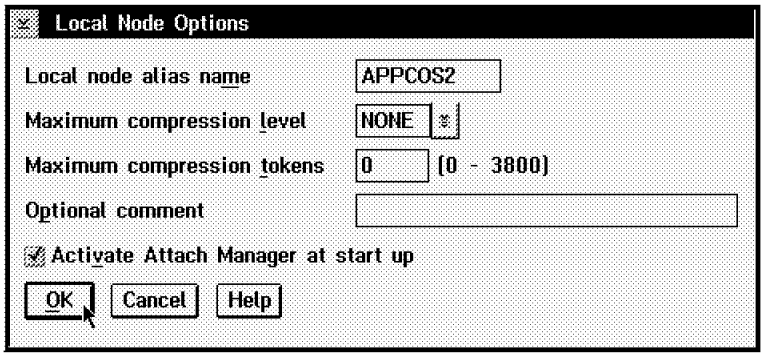


Figure 46. Alias to Local Node

In SNA connections we configure a peer-to-peer node to the DataJoiner machine, as shown in Figure 47 and Figure 48 on page 62.

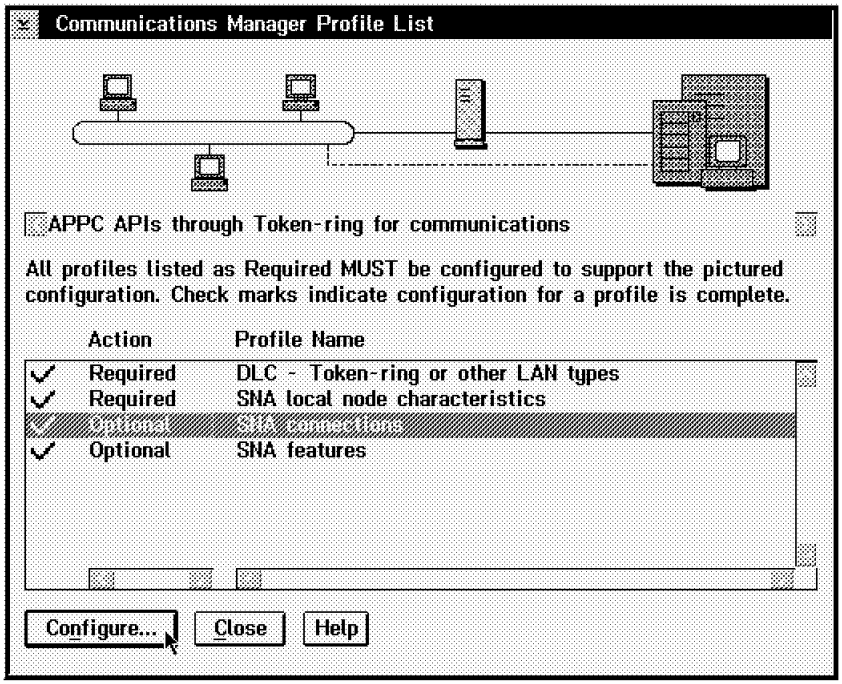


Figure 47. SNA Connections

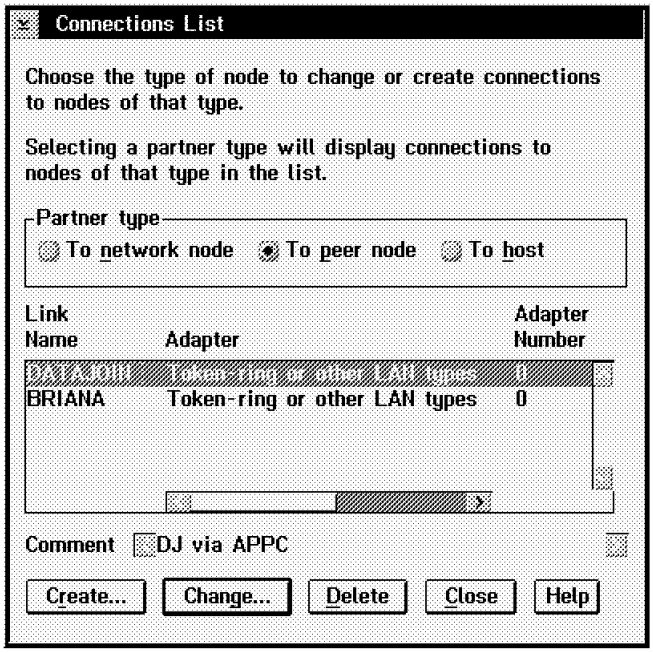


Figure 48. Connection to DataJoiner

We define a link name for the connection to DataJoiner. DATAJOIN is the link name we gave to this connection.

Figure 49 shows the window for selecting the LAN adapter type.

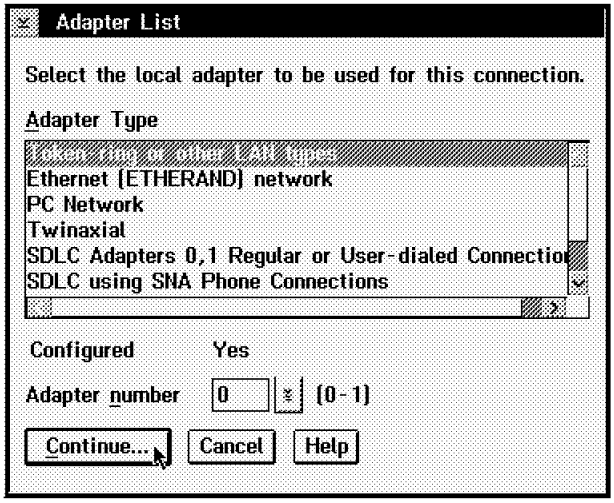


Figure 49. Selecting a LAN Adapter Type

Figure 50 on page 63 shows the window for connecting to a peer node.

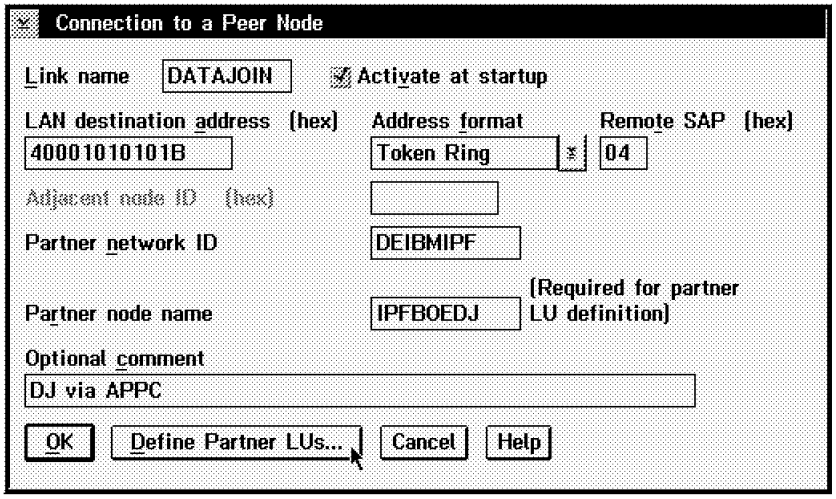


Figure 50. Connections to a Peer Node

The **LAN destination address** is the address in the LAN for the DataJoiner machine. The **Partner node name** designates the LAN Control Point in the SNA server configuration on the DataJoiner machine. The **Partner node name** must be the same as the LAN Control Point name; in the case of peer-to-peer connections, any name can be used.

Figure 51 shows the window for adding, changing, or deleting partner logical units.

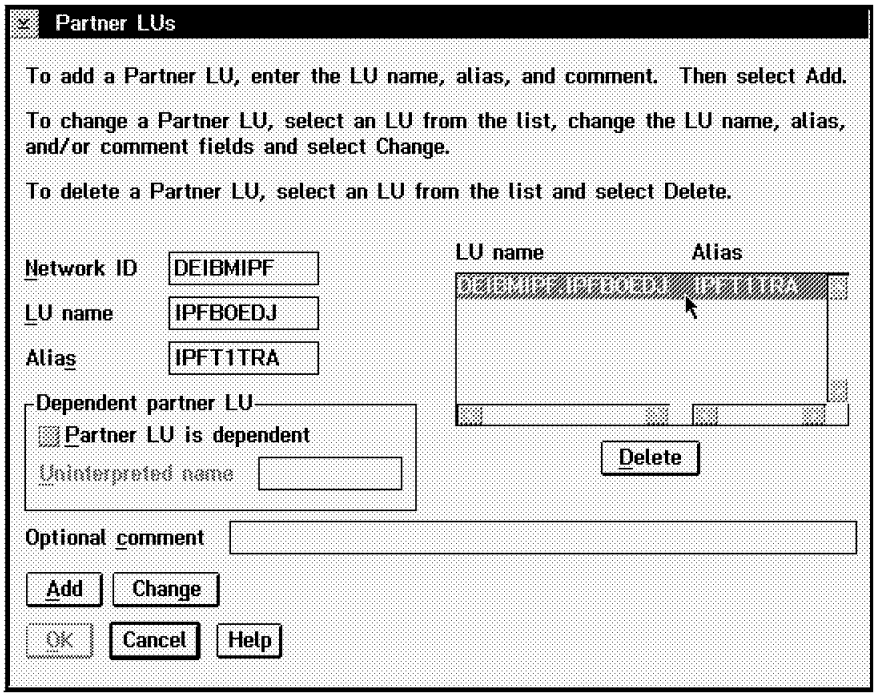


Figure 51. Partner LUs

The alias in Figure 51 is used to refer to the Network ID and LU name together.

Figure 52 on page 64 shows the window for the selection CM/2 profiles. These features include the parameters defined earlier for this example.

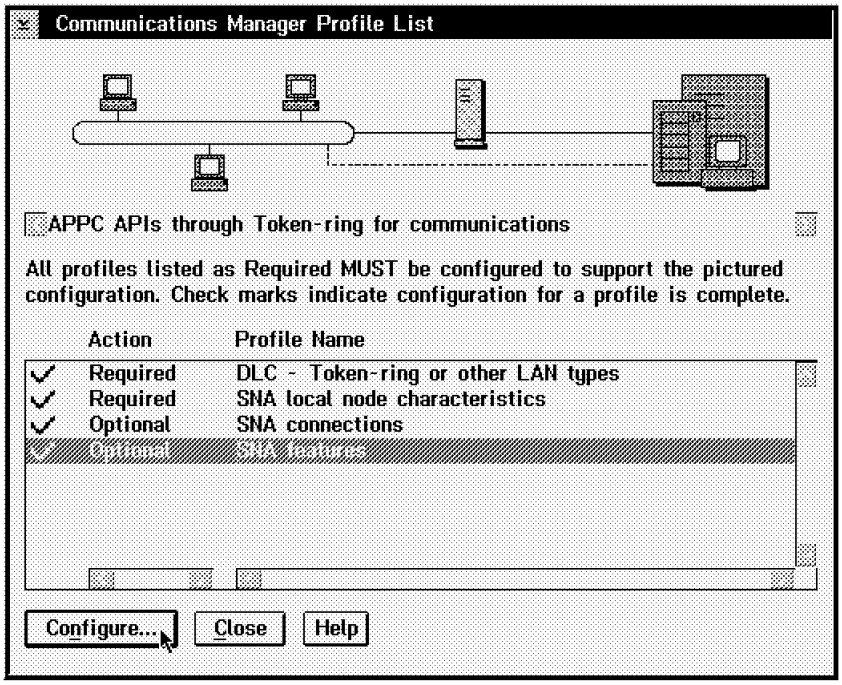


Figure 52. SNA Features

Figure 53 shows the SNA features, with a local logical unit selected.

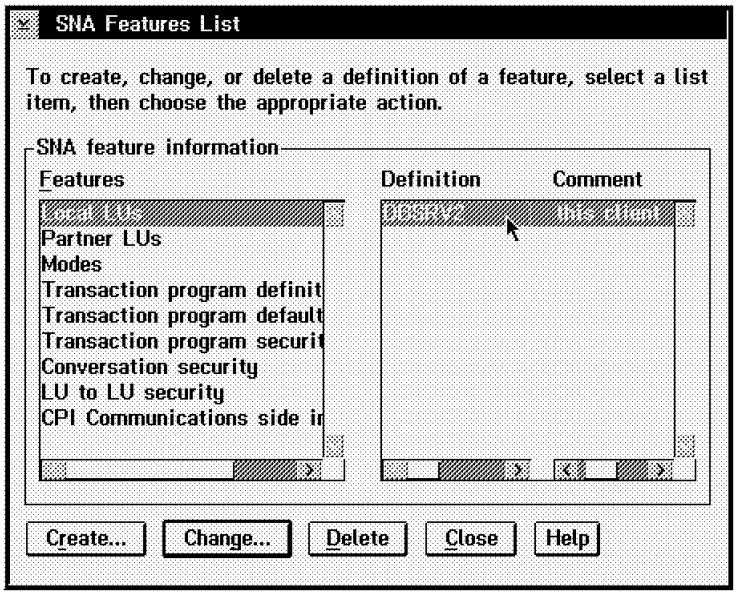


Figure 53. Selection of Local LU

The local LU name is the local node LU; we can use any name as an alias. The alias will be used by the local transaction programs (TPs) to access the local LU. See Figure 54 on page 65.

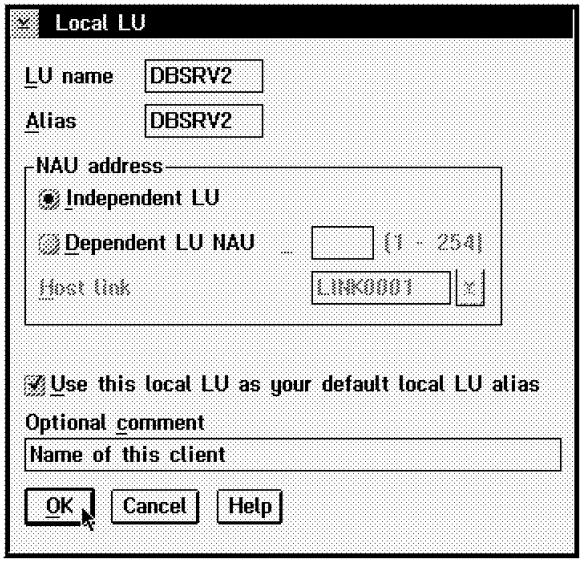


Figure 54. The Client Name

Figure 55 shows the SNA Features List with **Partner LUs** selected.

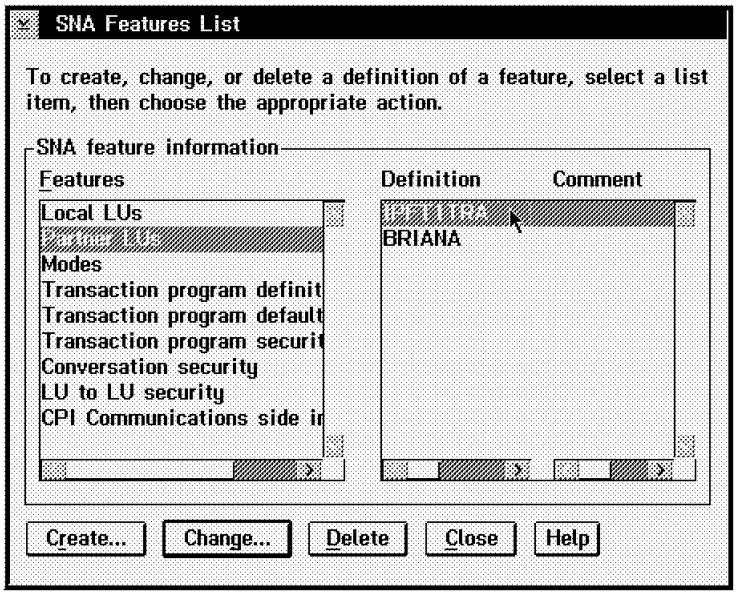


Figure 55. Partner LU

In this list, the task is to find the partner LU defined during peer-to-peer configuration. Figure 56 on page 66 shows the display of the partner LU information.

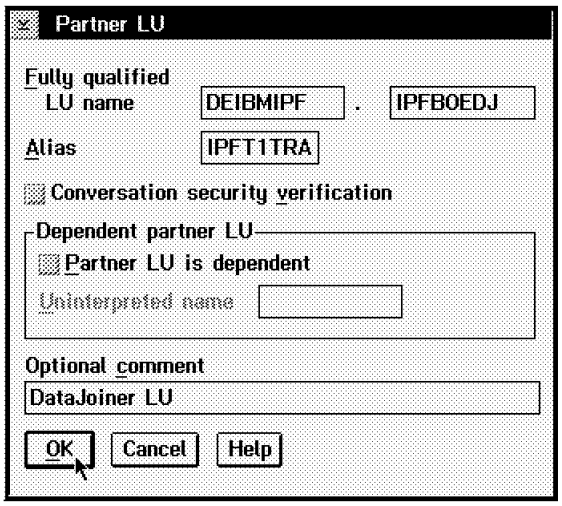


Figure 56. Partner Names

Figure 57 shows the SNA Features List window with **Modes** selected.

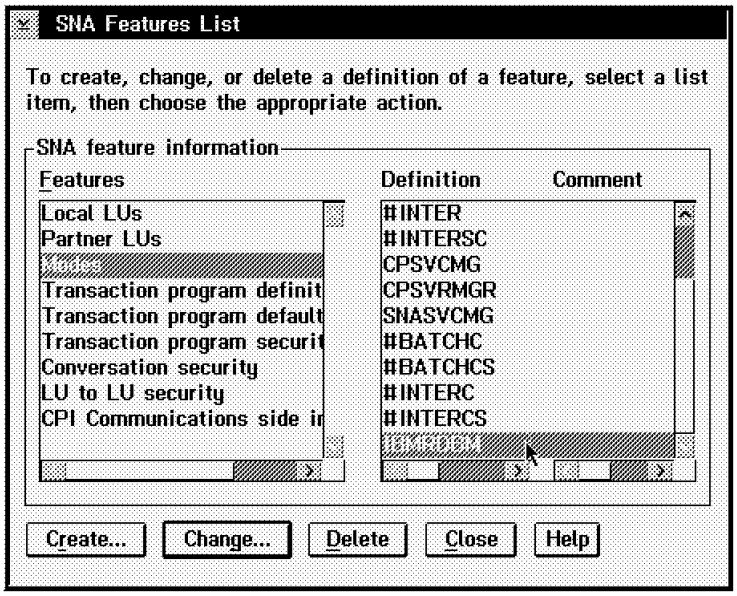


Figure 57. Mode Selection

For the mode specification, it is essential to use the same mode in the client and in the server (IBMRDBM) with its default parameters. See Figure 58 on page 67.

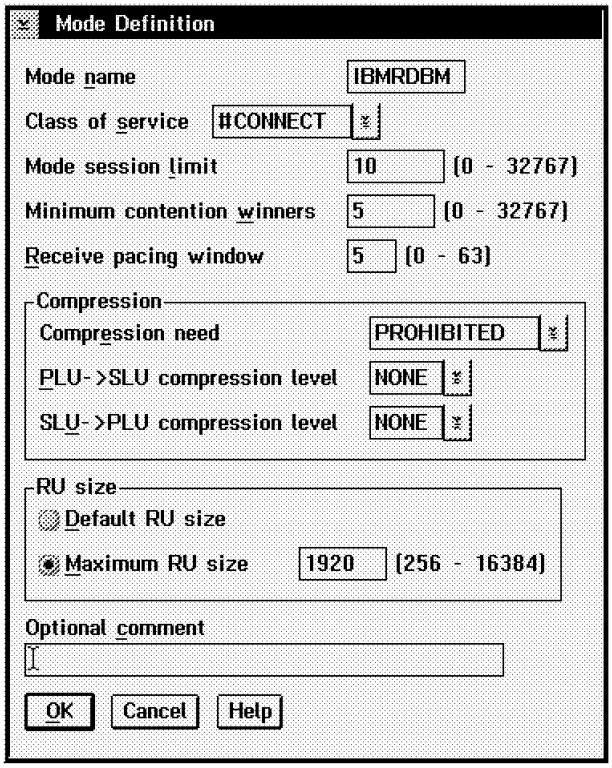


Figure 58. Mode Characteristics

In the CPI Communication Side Information window, choose a symbolic destination name, the alias for the partner LU, and the mode. Figure 59 and Figure 60 on page 68 show an example.

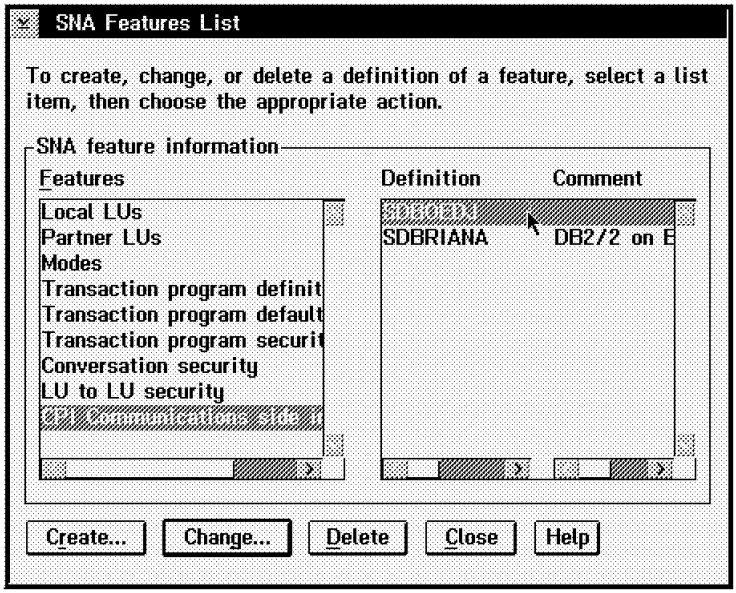


Figure 59. CPI Communications

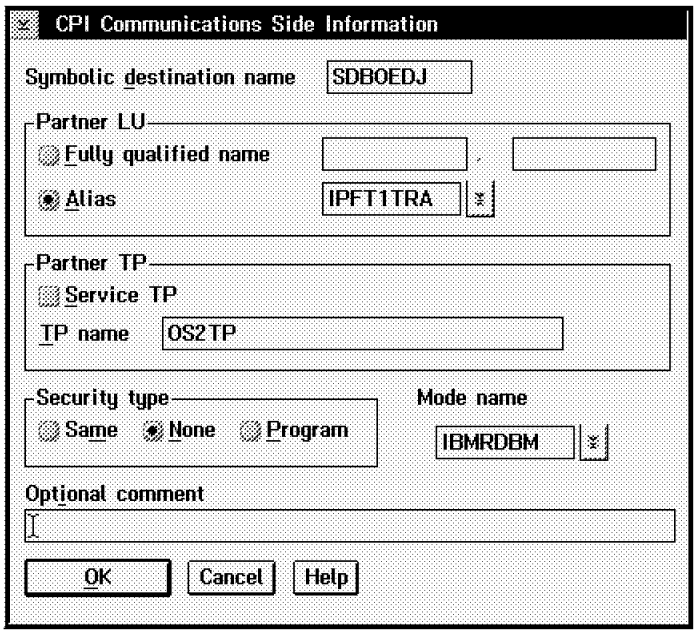


Figure 60. Side Information

For the transaction program name, we use the same program name on both sides (client and server). See 4.3, “Configuration of APPC Clients” on page 55. For security type see 6.2.4, “APPC Security” on page 148.

4.3.1.2 Update the System Catalog

To be able to connect to DataJoiner, you must catalog a node in the workstation directory. You can use the DB2/2 Directory Tool to do this. You must also catalog a database, and in some cases run a bind command. You can update the system database directory from the command line or use the DB2/2 Directory Tool within the DB2/2 group folder, if the client has DB2/2 installed.

Catalog a local node pointing to the remote database (DataJoiner) as in the following example.

```
catalog appc node nodedj remote sdboedj security xxxx
```

where

nodedj	Node entry name for use in the system database directory
sdboedj	Name of the LU 6.2 side information profile for this node
xxxx	See Chapter 6, “Security” on page 145 for information on this parameter.

For a description of the security parameter, see Chapter 6, “Security” on page 145. The default is NONE.

Catalog the remote database to the local database directory as in the following example.

```
catalog database datajoin as boedj at node nodedj authentication xxxxx
```

For a description of the parameters, see 4.2.1.2, “Update the System Catalog” on page 51. For a description of the authentication parameter, see Chapter 6, “Security” on page 145. The default is SERVER.

Before you can use DataJoiner, known in the client as boedj, you must connect to it.

To connect successfully to DataJoiner, you need a user ID and password in the AIX system that was defined to access DataJoiner. You get an SQL error message (-1403) if the ID and password are not correct. See 6.2.7, “Passwords at DataJoiner Server” on page 149. Here is an example of a CONNECT statement:

```
connect to boedj user userid using password
```

After the CONNECT statement, under certain conditions you must run a bind command. Each time you create a new database on your server, packages for the database utilities must be created in the catalog tables of that database′s system. These packages can be created by running the bind command once from each type of client, for example CAE/2 Version 1 clients, CAE/2 Version 2 clients, DB2/2 Version 1 clients, and DB2/2 Version 2 clients. Here is an example of the bind command:

```
bind \sqllib\bnd\@db2ubind.lst blocking all
```

Note: DB2/2 Version 1 and CAE/2 Version 1 cannot reside on the same partition.

4.3.2 Configuration of AIX Clients

If you have DB2/6000 clients installed, you have already connected these clients with CAE/6000 to an installed DB2/6000 system. To configure one of these clients for connection to DataJoiner, you must take similar steps.

Before starting on the configuration, you should have SNA Server/6000 installed.

To carry out this configuration, you need root authority.

After you login as root, you need to

1. Configure SNA Server/6000
2. Update the system database.

4.3.2.1 Configure SNA Server/6000

In SNA Server/6000, you must create profiles for

- Initial node setup
- Token-ring link station
- Local LU
- Side information
- Mode definition
- Partner LU location.

Initial Node Setup: During installation of SNA Server/6000, it is likely that the initial node was set up. If this is the case, do not repeat the setup procedure. If the initial node was not set up at installation time, issue the command `smit sna` to enter information into the SNA Server/6000 configuration window.

Select the following options from the menu to go to the node setup window:

Configure SNA Profiles
Initial Node Setup.

Figure 61 shows the settings for our example.

```
                              Initial Node Setup

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

                                                      [Entry Fields]
  Control Point name                          +       [IPFP221C]
  Control Point type                                   appn_end_node
  Local network name                                  [DEIBMIPF]
  XID node ID                                         [*]

  Optional link station information:

  Link station type                                    token_ring
  Link station name                                   []
* Calling link station?                               yes
  Link address                                        []
```

Figure 61. LAN Control Point Profile

Define the following fields:

Control point name. Any name you want; it can be the PU name.

Control point type Must be appn_end_node

Local network name The name for your local network. In our example it is DEIBMIPF.

XID node ID In our example, these values are 071 and E001C, respectively.

This ID can be set here or in the Token Ring Link Station Profile.

Leave the other fields with the default values, and press Enter to save the profile.

Configure a Token-Ring Link-Station Profile: Exit from the initial node setup window and return to the Configure SNA Profiles menu. Select the following options to go to the token-ring link station window illustrated in Figure 62:

Configure SNA Profiles
Advanced Configuration
Links
Token Ring
Token Ring Link Station
Add a Profile.

```
                    Add Token Ring Link Station Profile

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

[TOP]                                                  [Entry Fields]
  Profile name                                        [appcdjbb]
  Use Control Point's XID node ID?                     no
     If no, XID node ID                               [071E001C]
* SNA DLC Profile name                                [tok0.00001]
  Stop link station on inactivity?                     no
     ....
  Adjacent Node Address Parameters
     Access routing                                    link_address
     If link_address,
        Remote link address                           [40001010101B]
        Remote SAP address (02-fa)                    [04]

  Adjacent Node Identification Parameters
     Verify adjacent node?                             no
     Network ID of adjacent node                      [DEIBMIPF]
     CP name of adjacent node                         [DEIBMIPF]
     XID node ID of adjacent node (LEN node only)     [*]
     Node type of adjacent node                        appn_end_node
     ....
```

Figure 62. Token-Ring Link-Station Profile

Define the following fields:

Profile name Any name you want

XID node ID In our example, these values are 071 and E001C, respectively.

This ID can be set here or in the Initial Node Setup.

SNA DLC profile Press F4 on this field and select tok0.0001

Remote link address Token ring address of the server (RISC System/6000)

Network ID of the adjacent node
Network name on which the server is connected

Leave the other fields with the default values, and press Enter to save the profile.

Configure the Local LU: Return to the Configure SNA Profiles menu, and select the following options:

Advanced Configuration
Sessions
LU 6.2
LU 6.2 Local LU.

You can choose to add or change a profile, depending on whether you have already defined this profile.

Figure 63 shows the definition for our example.

```
                    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                               appclu01
  New profile name                                   []
  Local LU name                                      [IPFCL01C]
  Local LU alias                                     [IPFCL01C]
  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                                           []
```

Figure 63. Local LU Profile Definitions

Define the following fields:

Profile name Use any name you want; it can be the LU name.

Local LU name LU name that you want to assign to the system. This LU must be defined in VTAM under the PU specified in the Initial Node (Control Point)—IPFP221C in our example (from Figure 61 on page 70).

Local LU alias Alternative name for the local LU. Our example uses the local LU name.

Local LU is dependent?
Must be set to no

Leave the other fields with the default values, and press Enter to save the profile.

Configure a LU 6.2 Side Information Profile: Exit from the LU 6.2 local LU profile window, and return to the LU 6.2 menu. Select the following options:

LU 6.2 Side Information
Add a profile.

Figure 64 shows how we define the side information profile for our example.

```
                    Add LU 6.2 Side Information Profile

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

                                                      [Entry Fields]
* Profile name                                       [itso1]
  Local LU or Control Point alias                    [IPFCLO1C]
  Provide only one of the following:
     Partner LU alias                                []
     Fully qualified partner LU name                 [DEIBMIPF.IPFBOEDJ]
  Mode name                                          [IBMRDBM]
  Remote transaction program name (RTPN)             [os2tp]
  RTPN in hexadecimal?                                no

  Comments                                           [DJ in itso1]
```

Figure 64. Side Information Profile in Germany to Access DataJoiner

Define the following fields:

Profile name Any name you want

Local LU or Control Point alias
The name of the local LU defined before. See Figure 63 on page 72

Fully qualified partner LU name
The name of the network and the LU name of the partner system. In our case, this system is itso1.

This information is in the initial-node setup window (see Figure 61 on page 70) and in the LU 6.2 local LU profile (see Figure 63 on page 72).

Mode name Must match the LU 6.2 mode profiles defined in the server machine (DataJoiner on itso1).

Remote transaction program name (RTPN)
Must match the first transaction program name profile defined in DataJoiner, which is zzservertp. For this program, developers defined an alias os2tp, so that we can use the alias as well.

Leave the other fields with the default values, and press Enter to save the profile.

Configure a LU 6.2 Mode Profile: Exit from the LU 6.2 local LU window, return to the LU 6.2 menu, and select the following option to go to the LU 6.2 mode menu:

LU 6.2 Mode

You can choose to add or change a profile, depending on whether you have already defined this profile. Figure 65 shows the definition for our example.

```
                        Change/Show LU 6.2 Mode Profile

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

                                                        [Entry Fields]
  Current profile name                                   IBMRDBM
  New profile name                                      []
  Mode name                                             [IBMRDBM]
  Maximum number of sessions (1-5000)                   [10]
  Minimum contention winners (0-5000)                   [5]
  Minimum contention losers (0-5000)                    [5]
  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)      [2816]
  Minimum RU size (128,...,32768: multiples of 32)      [1024]
  Class of Service (COS) name                           [#CONNECT]

  Comments                                              []
```

Figure 65. LU 6.2 Mode Profile

Define the following fields:

Profile name Can be any name; we use IBMRDBM for a relational database connection.

Mode name Name you assign to the mode. You can use IBMRDBM.

Leave the other fields with the default values, and press Enter to save the profile.

Configure a Partner LU 6.2 Location Profile: Exit from the LU 6.2 mode profile window, and return to the LU 6.2 menu. Select the following options:

Partner LU 6.2 Location

You can choose to add or change a profile depending on whether or not you have already defined this profile.

Figure 66 on page 75 shows the definition of the partner LU 6.2 location profile for our example.

```
                    Add Partner LU 6.2 Location Profile

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

                                                      [Entry Fields]
* Profile name                                       [ibmboedj]
  Fully qualified partner LU name                    [DEIBMIPF.IPFBOEDJ]
  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                                   [IPFCLO1C]
     Link Station Profile name                       [appcdjbb]

  Comments                                           [DJ in itso1]
```

Figure 66. Partner LU 6.2 Location Profile

Define the following fields:

Profile name Any name you want

Fully qualified partner LU name
The name of the network and the LU name of the partner system. In our case this system is itso1 and its network and LU name is DEIBMIPF.IPFBOEDJ.

This information is in the initial node setup window of itso1, and in the LU 6.2 local LU profile. See page 70 for the node specification, and page 72 for the local LU profile specification. Also see “Configure the Local LU 6.2” on page 130.

Partner LU location method
Set it to link_station

Local LU name Must match the name defined in the LU 6.2 local LU profile on page 72.

Link Station Profile name
Must match the token ring link station profile name defined on page 71.

Leave the other fields with the default values, and press Enter to save the profile.

Verify the configuration profiles: Exit from the partner LU 6.2 location profile window and press the F3 key until you reach the *Advanced Configuration menu*, then select:

Verify configuration profiles.

Set the update action field to dynamic_update with the tab key, and press Enter to verify the profiles. If the verification is not successful, return to the configuration menus to fix the problem.

Start the SNA Link Station: Exit from the Verify Configuration Profile window and press F3 repeatedly until you reach the SNA Server/6000 menu. Select the following options:

Manage SNA resources
Start SNA Resources
Start an SNA link station

In the window Start an SNA Link Station, press F4, and select from the list the name of the link station profile you defined on page 71. Press Enter to start the link station and leave SMIT.

4.3.2.2 Update System Catalog

These steps can be done with instance owner or system administrator (sysadm) authority.

You need to

- Catalog a CPIC node in the node directory.
- Catalog the database in the system database directory.

Catalog a CPIC Node in the Node Directory:

Login as the instance owner or as a user with system administrator (sysadm) authority, and catalog a CPIC node using the following command:

```
db2 catalog cpic node appcnod remote os2tp security sec
```

where

appcnod	Node entry name for use in the system database directory
os2tp	Transaction program name defined to access the server.
sec	Set it to `program` if the authentication type of the target database is set to `server`.

Catalog a Database in the System Database Directory:

This step is required only if you want to access this data source using the CONNECT TO SQL statment. Here is an example:

```
db2 catalog database datajoin as boedja at node appcnod authentication auth
```

where

datajoin	Name of a database under a DataJoiner instance.
boedja	Alternative name (alias) for the database being cataloged
appcnod	Specifies the node where the DataJoiner server resides. This must match the name you specify in the node directory. See “Catalog a CPIC Node in the Node Directory.”

auth The possible values are server, dcs, or client.

In our example, the database is created with autenthication set to server. (For security details see Chapter 6, “Security” on page 145.)

The client should now be able to successfully issue the CONNECT statement given a valid user ID and password to access DataJoiner. Here is an example of the CONNECT statement:

```
connect to boedja user userid using password
```

Chapter 5. Configuration of DataJoiner for Data Sources

The data sources for this project are these: DB2/MVS, SQL/DS, DB2/400, DB2/VSE, DB2/2, DB2/6000, Oracle, Sybase, and DataJoiner. They are configured for use with DataJoiner as described in the chapter sections listed in Table 3.

Table 3. Configuring Data Sources for Use with DataJoiner

Data Source	**Location of Configuration Instructions**
Sources configured using DRDA: DB2/MVS SQL/DS DB2/400 DB2/VSE	5.1, "Connecting DataJoiner to DRDA-Attached Data Sources" on page 80
Sources configured using JRA: DB2/6000 DataJoiner	5.2, "Connecting DataJoiner to Data Sources Using JRA" on page 98
DB2/2	5.2.4, "Configuring DB2/2 V2 as a Data Source" on page 114
Sybase	5.3.1, "Configuring Sybase Data Access Module" on page 137
Oracle	5.3.2, "Configuring Oracle Data Access Module" on page 141

Figure 67 on page 80 represents these data sources graphically. The steps to configure each data source are described in detail in the sections that follow.

The names "Yellow," "Severn," and "Zeus" are the TCP/IP host names for the AIX/6000 systems used in the implementation.

© Copyright IBM Corp. 1995

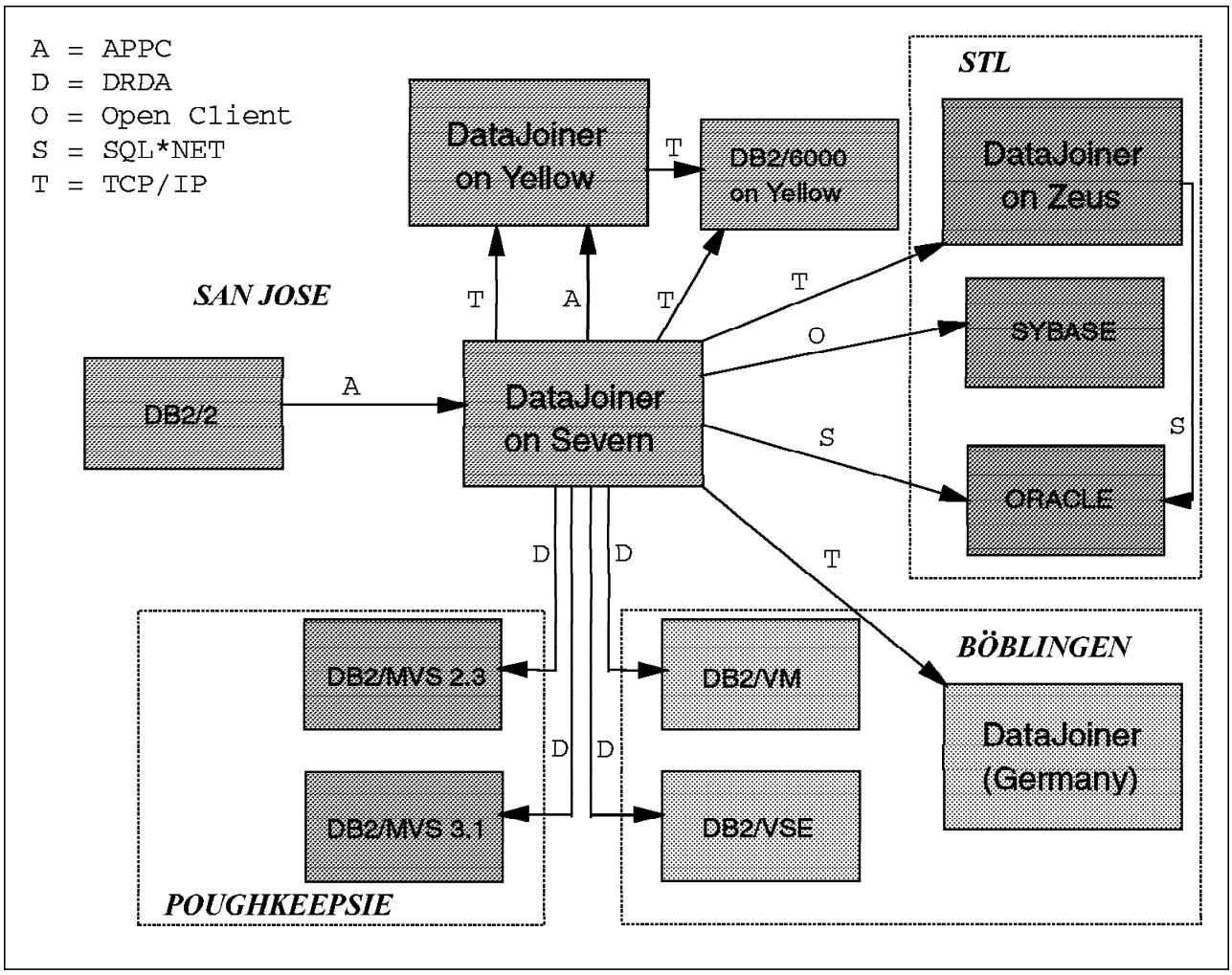


Figure 67. DataJoiner Data Sources

5.1 Connecting DataJoiner to DRDA-Attached Data Sources

This section describes the steps required to connect DataJoiner to DRDA-attached data sources. Data sources that DataJoiner accesses using DRDA include:

- DB2/MVS
- SQL/DS
- DB2/VSE
- DB2/400.

To connect DataJoiner to any DRDA-attached data source, you need to:

- Configure the SNA Server/6000 V2 to access the data source.
- Catalog the CPIC node.
- Bind the DataJoiner packages to the remote DRDA data source.
- Update the System Catalog Tables.

When you have completed these steps, see 7.3.4, “Create Nicknames for Remote Tables and Views” on page 165 for information on creating nicknames for the tables you wish to access in the data source.

5.1.1 Configure SNA Server/6000 V2 to Access the Data Source

DRDA-attached data sources are accessed by DataJoiner using SNA LU 6.2 protocols. SNA Server/6000 V2 must be installed on the DataJoiner machine before DataJoiner can access these sources.

To configure SNA Server/6000 V2 to access a DRDA data source, you need to complete the following steps or verify that each has been completed:

1. Perform Initial node setup.
2. Configure the Link station.
3. Configure an LU 6.2 Local LU profile.
4. Configure an LU 6.2 Side Information profile.
5. Configure an LU 6.2 Mode profile.
6. Configure an LU 6.2 Partner LU Location profile.
7. Verify the configuration profiles.

Each step is described in detail in the sections that follow.

Use SMIT to configure SNA Server/6000 V2 LU 6.2 profiles. Login as root and enter

```
smit sna
```

at the command line to see the SNA Server/6000 V2 window.

5.1.1.1 Initial Node Setup

If this is a new installation of SNA Server/6000 V2 on the DataJoiner machine, you must perform Initial Node Setup.

Select the following options from the SNA Server/6000 V2 window:

1. **Configure SNA Profiles**
2. **Initial Node Setup.**

Before you see the Initial Node Setup window, you will be prompted to select the correct data link control (DLC) type for your DataJoiner installation. You may use PF4=List to see the list of valid options. Select a DLC type and press Enter. We selected `token_ring` as our DLC type. Figure 68 on page 82 shows the second Initial Node Setup window.

```
                              Initial Node Setup

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

                                                        [Entry Fields]
  Control Point name token_ring                +       [SCA2085]
  Control Point type                                    appn_end_node
  Local network name                                    [USIBMSC]
  XID node ID                                           [071a2085]

  Optional link station information:

  Link station type                                     token_ring
  Link station name                                     []
* Calling link station?                                 yes
  Link address                                          []




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
```

Figure 68. Initial Node Setup

Update the following fields with the values for your DataJoiner installation:

Control Point name Choose any name. Our example uses SCA2085.

Control Point type Use *appn_end_node*, unless in an APPN network, when you should use *appn_network_node*.

Local network name This must match the VTAM Startup List NETID parameter in VTAM. Our example uses USIBMSC.

XID node ID This value must match the IDBLK= and IDNUM= values on the unique PU definition for this AIX machine in VTAM. Ask your VTAM administrator for the correct value for your machine. Our example uses 071a2085. See Appendix D, "MVS VTAM Definitions in San Jose" on page 221 for a listing of the PU and LU definitions in VTAM.

Use defaults for the rest of the fields. Press Enter to save the profile information.

If Initial Node Setup has already been completed, verify that all the values are correct.

5.1.1.2 Links

If this is a new installation of SNA Server/6000 V2 on the DataJoiner machine, you need to add a Token Ring Link Station Profile for the VTAM host. Select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Links.

From this window, select the correct network type for your environment. We selected Token Ring. If you use another network type, you may have to select different options to reach the Add a Link Station window.
Token Ring
Token Ring Link Station
Add a Profile.

Figure 69 shows the first part of the Add Token Ring Link Station Profile window.

```
                  Add Token Ring Link Station Profile

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

[TOP]                                  [Entry Fields]
* Profile name                                        [trlink]
  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, LU Address Registration Profile name      []
  Trace link?                                          no
     If yes, Trace size                                long

  Adjacent Node Address Parameters
     Access routing                                    link_address
[MORE...38]

F1=Help            F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset        F6=Command          F7=Edit             F8=Image
F9=Shell           F10=Exit            Enter=Do
```

Figure 69. Add Token Ring Link Station Profile (1 of 2)

Update the following fields with the values for your DataJoiner installation:

Profile name
Make up a descriptive name. Our example uses

```
trlink
```

.

SNA DLC Profile name
Use PF4=List to see the list of already defined SNA DLC profiles. Select a profile name from this list. We use the default SNA DLC profile called

```
tok0.00001
```

.

Scroll down this window to see the rest of the fields. Figure 70 on page 84 shows some of the second part of the Add Token Ring Link Station Profile window.

```
                  Add Token Ring Link Station Profile

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

[MORE...12]                                            [Entry Fields]
  Adjacent Node Address Parameters
     Access routing                                     link_address
     If link_name, Remote link name                    []
     If link_address,
        Remote link address                            [400008210200]
        Remote SAP address (02-fa)                     [04]


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 70. Add Token Ring Link Station Profile (2 of 2)

Adjacent Node Address Parameters

Use the value

`link_address`

Remote link address

Specify the token-ring address of the destination host. In our case, this is the token-ring interface card (TIC) address of our communications controller and has the value

400008210200

.

Use defaults for the rest of the fields. Press Enter to save the profile information.

If a Token Ring Link Station Profile already exists, select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Links.

From this window, select the correct network type for your environment. We selected Token Ring. If you use another network type, you may have to select different options to reach the window for the Add Token Ring Link Station Profile.
Token Ring
Token Ring Link Station
Change/Show a Profile.

From this window, select the name of the existing profile you want to use. **PF4=List** will show you a list of the valid options. Check that all the fields contain the correct values.

5.1.1.3 Sessions

DRDA communication requires an LU 6.2 Local LU profile to be defined once for the DataJoiner machine. For each DRDA-attached data source, you must define an LU 6.2 Side Information profile, an LU 6.2 Mode profile, and a Partner LU 6.2 Location profile.

LU 6.2 Local LU: Select the following options from the SNA Server/6000 V2 window to create an LU 6.2 Local LU Profile:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Local LU
Add a Profile.

Figure 71 shows the Add LU 6.2 Local LU Profile window.

```
                        Add LU 6.2 Local LU Profile

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

                                                      [Entry Fields]
  Profile name                                       [SCA2085I]
  Local LU name                                      [SCA2085I]
  Local LU alias                                     [SCA2085I]
  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
Esc+5=Reset        F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

Figure 71. Add LU 6.2 Local LU Profile

Update the following fields with the values for your DataJoiner Installation:

Profile name — Choose any descriptive name. Our example uses `SCA2085I`

Local LU name — Specify a unique independent LU name defined for the DataJoiner machine in VTAM. Our example uses `SCA2085I`
See Appendix D, “MVS VTAM Definitions in San Jose” on page 221 for a listing of our VTAM PU and LU definitions.

Local LU alias	Choose any alias name. Our example uses `SCA2085I`
Local LU is dependent?	Make sure that this field is set to `no`

Use defaults for the rest of the fields. Press Enter to save the profile information.

If an LU 6.2 Local LU definition already exists, verify that all the values defined in it are correct. View the profile by selecting the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Local LU
Change/Show a Profile.

From this window, select the name of the existing profile you want to use. **PF4=List** will show you a list of the valid options. Check that all the fields contain the correct values.

LU 6.2 Side Information: A unique LU 6.2 Side Information profile that points to the partner LU is required for every DRDA data source. This profile is also used to catalog the node in the Node Directory. To add a profile, select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Side Information
Add a Profile.

Figure 72 on page 87 shows the Add LU 6.2 Side Information Profile window.

```
                    Add LU 6.2 Side Information Profile

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

                                                      [Entry Fields]
  Profile name                                       [DB2VMTPN]
  Local LU or Control Point alias                    [SCA2085I]
  Provide only one of the following:
     Partner LU alias                                []
     Fully qualified partner LU name                 [DEIBMIPF.IPFA2GL4]
  Mode name                                          [IBMRDBM]
  Remote transaction program name (RTPN)             [S34VMDBO]
  RTPN in hexadecimal?                                no

  Comments                                           [SQL/DS GERMANY]




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 72. Add LU 6.2 Side Information Profile

Update the following fields with the values for your DataJoiner Installation:

Profile name
Choose any descriptive name. Our example uses

DB2VMTPN

and is for connecting DataJoiner to a SQL/DS system.

Local LU or Control Point alias
Use the alias defined in “LU 6.2 Local LU” on page 85. Our example uses

SCA2085I

.

Partner LU alias
If an LU 6.2 Partner LU profile is already defined for the data source, you can use its alias name as the Partner LU alias. If you have not defined a Partner LU Profile, use the Fully qualified partner LU name instead.

Fully qualified partner LU name
Specify *networkname.partnerluname* for the DRDA data source. The *networkname* is the name of the network where the DRDA data source LU is defined. The *partnerluname* is the name on the APPL statement in VTAM for the DRDA data source. Our example uses the value

DEIBMIPF.IPFA2GL4

Appendix C, “VM/ESA Definitions in Germany” on page 215 contains a listing of the VM APPL definitions.

Mode name

Specify the mode name defined in the LU 6.2 mode “LU 6.2 Mode.” Our example uses

```
IBMRDBM
```

Remote Transaction Program Name (RTPN)

For DB2/MVS or DB2/400, this value is

```
X'07F6C4C2'
```

For SQL/DS or DB2/VSE the TPN value is usually the *VM RESID*.

RTPN in hexadecimal?

If the data source is SQL/DS or DB2/400, specify

```
yes
```

For DB2/MVS and DB2/VSE, specify

```
no
```

.

Press Enter to save the profile information.

If an LU 6.2 Side Information profile already exists, verify that all the values defined in it are correct. View the profile by selecting the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Side Information
Change/Show a Profile.

From this window, select the name of the existing profile you want to use. **PF4=List** will show you a list of the valid options. Check that all the fields contain the correct values.

LU 6.2 Mode: To define an LU 6.2 Mode profile, select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Mode
Add a Profile.

Figure 73 on page 89 shows the Add LU 6.2 Mode profile window.

```
                        Add LU 6.2 Mode Profile

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

                                                        [Entry Fields]
  Profile name                                         [IBMRDBM]
  Mode name                                            [IBMRDBM]
  Maximum number of sessions (1-5000)                  [10]
  Minimum contention winners (0-5000)                  [5]
  Minimum contention losers (0-5000)                   [5]
  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)     [2816]
  Minimum RU size (128,...,32768: multiples of 32)     [1024]
  Class of Service (COS) name                          [#CONNECT]

  Comments                                             []


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 73. Add LU 6.2 Mode Profile

Update the following fields with the values for your DataJoiner Installation:

Profile name Choose any descriptive name. Our example uses the value

IBMRDBM

Mode name This name must be defined as the MODEENT name in the mode table of the local VTAM host. If the DRDA data source has a different local VTAM host, it must be defined there also. Our example uses the value
IBMRDBM
Its definition is listed in Appendix D, “MVS VTAM Definitions in San Jose” on page 221.

For performance reasons, you may want to customize the maximum number of sessions defined for your data source. Please refer to 8.1.1, “LU 6.2 Sessions and Related DB2 Parameters” on page 175, for detailed information on customizing this parameter. Accept the defaults for the remaining values.

Press Enter to save the profile information.

If an LU 6.2 Mode Profile already exists for your DRDA data source, verify that all the values defined in it are correct. View the profile by selecting the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
LU 6.2 Mode

Change/Show a Profile.

From this window, select the name of the existing profile you want to use. **PF4=List** will show you a list of the valid options. Check that all the fields contain the correct values.

Partner LU 6.2 Location: A unique Partner LU 6.2 Location profile that connects the remote LU and the local LU is required for every DRDA data source. To add a partner LU 6.2 location profile, select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
Partner LU 6.2 Location
Add a Profile.

Figure 74 shows the Add Partner LU 6.2 Location Profile window.

```
                    Add Partner LU 6.2 Location Profile

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

                                                      [Entry Fields]
  Profile name                                       [SQLDSPLU]
  Fully qualified partner LU name                    [DEIBMIPF.IPFA2GL4]
  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                                   [SCA2085I]
     Link Station Profile name                       [trlink]

  Comments                                           [SQL/DS GERMANY]



F1=Help            F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset        F6=Command          F7=Edit             F8=Image
F9=Shell           F10=Exit            Enter=Do
```

Figure 74. Add Partner LU 6.2 Location Profile

Update the following fields with the values for your DataJoiner Installation:

Profile name — Choose any descriptive name. Our example uses SQLDSPLU

Fully qualified partner LU name — Specify *networkname.partnerluname* for the DRDA data dource. The *networkname* is the name of the network where the DRDA data source LU is defined. The *partnerluname* is the

	name on the APPL statement in VTAM for the DRDA data source. Our example uses the value `DEIBMIPF.IPFA2GL4` Our APPL definition is listed in Appendix C, "VM/ESA Definitions in Germany" on page 215.
Partner LU location method	Specify `link_station`
Local LU name	This is the local LU name defined in "LU 6.2 Local LU" on page 85.
Link Station Profile name	This is the name of the link station profile defined in 5.1.1.2, "Links" on page 82.

Press Enter to save the profile information.

If a Partner LU 6.2 Location Profile already exists for your DRDA data source, verify that all the values defined in it are correct. View the profile by selecting the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Sessions
LU 6.2
Partner LU 6.2 Location
Change/Show a Profile.

From this window, select the name of the existing profile you want to use. **PF4=List** will show you a list of the valid options. Check that all the fields contain the correct values.

5.1.1.4 Verify Configuration Profiles

Before your configuration changes can take effect, you must verify the configuration profiles and update SNA.

Select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles
Advanced Configuration
Verify Configuration Profiles.

Verify Configuration Profiles while SNA is Running: To verify your configuration profiles while SNA is running, change the `Update action if verification successful` field to **dynamic_update** and press Enter.

If there are any verification errors, correct them and verify the configuration profiles again.

If there are no errors, the following message is displayed:

```
verifysna command OK.
The profiles listed above have been dynamically updated successfully.
```

Your new configuration profiles are now ready to use.

As an alternative to selecting **dynamic_update**, you can verify the profiles by changing the Update action if verification successful field to **none** and pressing Enter.

If there are errors, correct them and verify the configuration profiles again. If there are no errors, the following message is displayed:

```
verifysna command OK.
No profiles updated. Use verifysna's update option to
commit profile modifications.
```

To make the new profiles available to SNA Server/6000 V2, you must stop and restart SNA. To stop SNA, select the following from the SNA Server/6000 V2 window:

Manage SNA Resources
Stop SNA Resources
Stop SNA.

On the Stop SNA panel, set Type of stop to perform to **normal**.

Press Enter on this panel and wait for the following message:

```
0513-044 The stop of the sna Subsystem was completed
successfully.
```

To restart SNA, select the following from the SNA Server/6000 V2 window:

Manage SNA Resources
Start SNA Resources
Start SNA

Press Enter on this panel. When SNA is started, the following message is displayed:

```
0513-059 The sna Subsystem has been started. Subsystem PID is
9142.
```

When SNA is started, press PF3 to return to the Start SNA Resources panel. From here, select
Start an SNA Link Station.

From this window select the name of the link station defined in 5.1.1.2, “Links” on page 82 and press Enter.

When the link is started, the following message is displayed:

```
0105-2723 The "trlink" Link Station has been started.
```

Now, your SNA Server/6000 V2 configuration profiles are ready to use.

Verify Configuration Profiles while SNA is Stopped: To verify your configuration profiles while SNA is stopped, change the Update action if verification successful field to **normal_update** and press Enter.

If there are any verification errors, correct them and verify the configuration profiles again.

If there are no errors, the following message is displayed:

```
verifysna command OK.
The profiles listed above have been dynamically updated successfully.
```

To start SNA, select the following from the SNA Server/6000 V2 window:

Manage SNA Resources
Start SNA Resources
Start SNA

Press Enter on this panel. When SNA is started, the follwing message is displayed:

```
verifysna command OK.
All profiles verified and updated.
```

When SNA is started, press PF3 to return to the Start SNA Resources panel. From here, select
Start an SNA Link Station.

From this window select the name of the link station defined in 5.1.1.2, “Links” on page 82 and press Enter.

When the link is started, the following message is displayed:

```
0105-2723 The "trlink" Link Station has been started.
```

Your SNA Server/6000 V2 configuration profiles are now ready to use.

5.1.2 Catalog CPIC Node

The command `catalog cpic node` adds information about the DRDA data source location to the DataJoiner database. It also provides the information required for DataJoiner to connect to the data source.

You must issue this command once for every DRDA data source.

You must have system administrator (`sysadm`) authority in the DataJoiner database to catalog a node.

Login as the instance owner and issue the following commands from the command line:

```
db2start
db2
connect to dbinst1
catalog cpic node nodename remote symbolic-destination-name security program
terminate
```

Substitute the following:

dbinst1	This is the name of your DataJoiner database instance.
nodename	Choose a unique and meaningful name. This must match the node name defined in SYSIBM.SYSSERVERS defined in 5.1.4.1, “Add an Entry to SYSIBM.SYSSERVERS” on page 97.
symbolic-destination-name	This is the name of the LU 6.2 Side Information Profile defined in “LU 6.2 Side Information” on page 86 for this node.

In our case, the commands entered were as follows:

```
db2start
db2
connect to dbinst1
catalog cpic node TOSQLDS remote DB2VMTPN security program
terminate
```

It is also possible to use a security option of same described in 6.2.4, “APPC Security” on page 148 in Chapter 6, “Security” on page 145.

5.1.3 Bind the DataJoiner Packages to the Remote DRDA Data Source

DataJoiner has packages that must be bound to each data source. If the packages have not already been bound at a data source, DataJoiner automatically attempts to bind them the first time access is attempted.

Since this bind occurs under the authorization ID of the first person to reference a data source, we recommend that this user ID have the correct authority at the data source for the bind to succeed.

If necessary, the DataJoiner bind can also be done manually.

The packages that are bound are the same as those used by the Distributed Database Connection Services (DDCS/6000) V1 utilities and the CAE Command Line Processor (CLP) to allow DRDA access to the data source. If either of these packages has already been bound to the data source, there is no need to bind it again.

To manually bind the DataJoiner packages to a data source, you must perform the following steps:

- Catalog the DRDA data source in the database connection services (DCS) directory.
- Catalog the DRDA data source in the system database directory.
- Bind the DataJoiner package at the DRDA data source.
- Clean up after successful bind.

5.1.3.1 Catalog the DRDA Data Source in the DCS Directory

This step is required only to perform the bind; it is not required to use DataJoiner.

This command maps a target database name to a location name (*RDB_NAME*).

You must have system administrator (sysadm) authority in the DataJoiner database to catalog a database.

Login as the instance owner and issue the following commands from the command line:

```
db2start
db2
connect to dbinst1
catalog dcs database database-name as target-database-name
```

Substitute the following:

dbinst1	This is the name of your DataJoiner database instance.
database-name	Choose a unique, meaningful name for the DCS directory. This must match the *database-name* used in the catalog system database command in 5.1.3.2, "Catalog the DRDA Data Source in the System Database Directory."
target-database-name	This is the DRDA data source location name (RDB_NAME). For example, in DB2/MVS this is the name found in the Boot Strap Data Set (BSDS).

In our case the commands entered were as follows:

```
db2start
db2
connect to dbinst1
catalog dcs database DBSQLDS as S34VMDBO
```

5.1.3.2 Catalog the DRDA Data Source in the System Database Directory

This step is required only to perform the bind; it is not required to use DataJoiner.

This command catalogs a remote database to the local DB2 instance.

You need to have system administrator (sysadm) authority in the DataJoiner database to catalog a database.

Login as the instance owner and issue the following command from the command line:

```
catalog database database-name as alias at node nodename authentication dcs
terminate
```

Substitute the following:

database-name	This is the name of this database as cataloged in the DCS directory in section 5.1.3.1, "Catalog the DRDA Data Source in the DCS Directory" on page 94.
alias	This is the name by which this database is known to DataJoiner. It is also the name that is used on the connect statement.
nodename	This is the node defined in the node directory for this DRDA data source in section 5.1.2, "Catalog CPIC Node" on page 93.

Authentication can also be specified as server or client. Refer to section 6.2.5, "Authentication Parameters" on page 148 in Chapter 6, "Security" on page 145 for further information on this option.

In our case, the command entered was as follows:

```
catalog database DBSQLDS as DBSQLDS at node TOSQLDS authentication dcs
```

5.1.3.3 Bind the DataJoiner Packages at the DRDA Data Source

Perform this step once for each DRDA data source.

You must have system administrator (sysadm) authority in the DataJoiner database to issue the db2start command.

If the DRDA data source is DB2/MVS, the user ID you use to perform the bind must have authority for, at least, BINDADD and CREATE IN COLLECTION NULLID.

If the DRDA data source is DB2/400, the administrator must create a collection called NULLID and ensure that the user ID executing the bind has access to the collection.

If the DRDA data source is DB2/MVS OR SQL/DS, the administrator must ensure that the user NULLID can perform DRDA connects and prepare and bind programs.

Login as the instance owner and issue the following commands from the command line:

```
db2 terminate
export DDCSSETP="-f=msg.out -s=e"
db2 connect to dbname user userid using password
db2 bind $HOME/sqllib/bnd/@ddcsbind.lst blocking all grant public
```

You must substitute the following:

dbname	This is the alias for the database as cataloged in the system database directory. See 5.1.3.2, "Catalog the DRDA Data Source in the System Database Directory" on page 95.
userid	This is the user ID that can perform the bind at the DRDA data source.
password	This is the password at the DRDA data source for user ID.

5.1.3.4 Cleanup after Successful Bind

Once the bind has completed successfully, remove the catalog entries and unset the DDCSSETP variable.

To test if the bind worked, issue the following command while still connected to the data source:

```
select name from SYSIBM.SYSTABLES where name='SYSTABLES'
```

To remove the database entries from the catalog (uncatalog them), we issued the following commands:

```
uncatalog database DBSQLDS
uncatalog dcs database DBSQLDS
```

```
db2 connect reset
db2 terminate
unset DDCSSETP
```

5.1.4 Update the System Catalog Tables

The system catalog consists of two new tables added to the relational database system catalog for DataJoiner use. The tables are created automatically by DataJoiner at installation time and are called SYSIBM.SYSSERVERS and SYSIBM.SYSREMOTEUSERS. The System Catalog contains information about the location of data sources and the remote-system user IDs and passwords needed to allow transparent data access.

5.1.4.1 Add an Entry to SYSIBM.SYSSERVERS

To add an entry to the SYSIBM.SYSSERVERS table, login to the DataJoiner machine as the instance owner and issue the following commands:

```
db2start
db2 connect to dbinst1
db2 "INSERT INTO SYSIBM.SYSSERVERS \
(SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
 VALUES ( 'server','node','dbname','type','version', 'protocol','password' )"
db2 connect reset
```

Substitute the following:

dbinst1 Specify the name of your DataJoiner database instance.

server Choose a descriptive name by which the data source will be known to DataJoiner. This field is defined as VARCHAR(18).

node Use the name defined in the node directory for this DRDA data source in 5.1.2, “Catalog CPIC Node” on page 93. This field is defined as VARCHAR(70).

dbname This is the name of the database at the data source. This is the same as the *target-database-name* in 5.1.3.1, “Catalog the DRDA Data Source in the DCS Directory” on page 94. This field is defined as VARCHAR(18).

type For DRDA data sources, this field must be DB2/MVS, DB2/VM, DB2/VSE, SQL/DS, or DB2/400.

version Specify the version of the database software. For example, if the source is DB2/MVS version 3.1, the value in the version field should be 3.1.

protocol For a DRDA source this field should always be drda. This field is case sensitive.

password Specify Y in this field so that the user ID and password are verified at the data source.

In our case the commands entered were as follows:

```
db2start
db2 connect to dbinst1
```

```
db2 "INSERT INTO SYSIBM.SYSSERVERS \
(SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
 VALUES('DB2VM','TOSQLDS','S34VMDBO','DB2/VM','3.4','drda','Y')"
db2 connect reset
```

5.1.4.2 Add an Entry to SYSIBM.SYSREMOTEUSERS

To add an entry to the SYSIBM.SYSREMOTEUSERS table, login to the DataJoiner machine as the instance owner and issue the following commands:

```
db2start
db2 connect to dbinst1
db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
 VALUES ('authid','server','remote_authid','remote_pw' )"
db2 connect reset
```

Substitute the following:

dbinst1 This is the name of your DataJoiner database instance.

authid This is the user ID on the DataJoiner system that will access the DataJoiner data source. This field is defined as char(8).

server This is the name of the data source as defined in 5.1.4.1, "Add an Entry to SYSIBM.SYSSERVERS" on page 97. This field is defined as VARCHAR(18).

remote_authid This is the user ID which is used at the data source. This field is defined as VARCHAR(30).

remote_pw This is the password used at the data source. This field is defined as VARCHAR(30).

In our case the commands entered were as follows:

```
db2start
db2 connect to dbinst1
db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
 VALUES ('DJINST1','DB2VM','IWDJO1','XXXXXX')"
db2 connect reset
```

Now you are ready to create nicknames for tables at the DRDA data source as described in 7.3.4, "Create Nicknames for Remote Tables and Views" on page 165. You can also use the DataJoiner passthru command now, to issue SQL statements to the data server that is connected to DataJoiner using that server's dialect of SQL. The passthru command is described in detail in the *DataJoiner Application Programming and SQL Reference Supplement*.

5.2 Connecting DataJoiner to Data Sources Using JRA

The JRA data access module is used to access DB2/2, DB2/6000 and DataJoiner data sources. However, DB2/2 must be Version 2 or higher and these data sources must have client-support capabilities. For example, DB2/6000 must have Client Support/6000 installed.

Figure 75 on page 99 shows the JRA data sources for which we describe the steps to establish the connections. These data sources are:

- DataJoiner, DB2/6000 V1 or DB2/6000 V2 in a remote machine using TCP/IP. See 5.2.1, “DataJoiner or DB2/6000 in a Remote Machine Using TCP/IP.”
- Other DataJoiner database in the same DataJoiner instance. See 5.2.2, “Other DataJoiner Database in the Same DataJoiner Instance” on page 106.
- Other DataJoiner, DB2/6000 V1 or DB2/6000 V2 instance in the same system. See 5.2.3, “Another DataJoiner Instance in the Same System” on page 110.
- DB2/2 V2. See 5.2.4, “Configuring DB2/2 V2 as a Data Source” on page 114.
- DataJoiner or DB2/6000 V1 in a remote machine using APPC. See 5.2.5, “DataJoiner or DB2/6000 V1 in a Remote Machine Using APPC” on page 122.

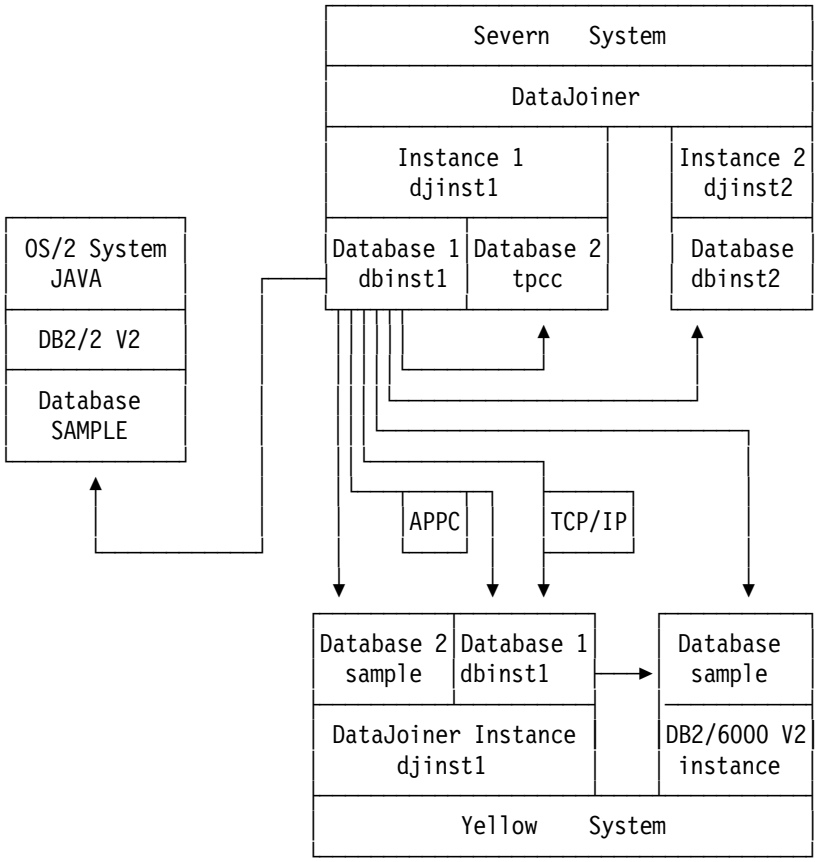


Figure 75. JRA Scenario. Shows the connection for the JRA data sources using TCP/IP, except for one connection from DataJoiner in the Severn system to the database called “dbinst1” in the Yellow system, for which we used APPC.

5.2.1 DataJoiner or DB2/6000 in a Remote Machine Using TCP/IP

The database manager capabilities of DataJoiner are based on the DB2/6000 V1 code. In fact, DataJoiner supports the same SQL dialect as DB2/6000 V1 with a few additions. These additions are for creation of nicknames for a table on a remote data source, or for use of the DataJoiner passthru function.

From the point of view of connectivity, connecting DataJoiner is the same whether to a data source in another DataJoiner system, in DB2/6000 V1, or in DB2/6000 V2. In this section, we show the connection to a remote DataJoiner, but we mention the differences pertinent to DB2/6000 V1 or V2 where they apply.

The steps required to connect DataJoiner to a remote DataJoiner or DB2/6000 data source through TCP/IP are these:

1. On the data-source side, configure DataJoiner for TCP/IP remote clients:
    a. Define the TCP/IP listen ports for DataJoiner.
    b. Refresh the inetd daemon.
    c. Change the database manager configuration.
    d. Set the DB2COMM environment variable to enable TCP/IP.
    e. Start the DataJoiner instance.
2. On the client side, configure DataJoiner to access a remote TCP/IP server:
    a. Define the DataJoiner server host name and location to AIX.
    b. Define the TCP/IP service ports to access the server.
    c. Refresh the inetd daemon.
    d. Catalog a node entry pointing to the location of the remote database.
    e. Catalog the remote database in the system database directory.
    f. Populate the System Catalog Tables for the DataJoiner data source.

To perform these steps, we assume that the minimum TCP/IP configuration is set; that is, you have a host name, a TCP address, and a domain name.

For the example used in this chapter, Figure 76 on page 101 shows the minimum TCP/IP configuration for the Severn system. The client DataJoiner runs on the Severn system. To get to this window, login as root in the Severn system, issue the command `smit`, and make the following selections from the menus:

Communications Applications and Services

TCP/IP

Minimum Configuration & Startup.

```
                        Minimum Configuration & Startup

 To Delete existing configuration data, please use Further Configuration


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

                                                        [Entry Fields]
* HOSTNAME                                             [severn]
* Internet ADDRESS (dotted decimal)                    [9.113.36.69]
  Network MASK (dotted decimal)                        [255.255.254.0]
* Network INTERFACE                                     tr0
  NAMESERVER
           Internet ADDRESS (dotted decimal)           [9.113.42.250]
           DOMAIN Name                                 [sanjose.ibm.com]
  Default GATEWAY Address                              [9.113.36.254]
          (dotted decimal or symbolic name)
  RING Speed                                           4
  START TCP/IP daemons Now                             no
```

Figure 76. Minimum TCP/IP Configuration Window

5.2.1.1 Configuring DataJoiner or DB2/6000 for TCP/IP Remote Clients

In this example we consider the Yellow system to be the data source. The steps to configure this DataJoiner for access by TCP/IP clients follow:

1. Define the TCP/IP listen ports for DataJoiner.
2. Refresh the inetd daemon.
3. Change the database manager configuration.
4. Set the DB2COMM environment variable to enable TCP/IP.
5. Start the DataJoiner instance.

Define the TCP/IP Listen Ports for DataJoiner: Login as root in the data source system and edit the /etc/services file using the vi editor. Add two entries like these to the file:

```
djmain          4800/tcp     # DataJoiner main connection port
djmaini         4801/tcp     # DataJoiner interrupt port
```

where “djmain” and “djmaini” are any names not already in the /etc/services file. The numbers for the ports 4800 and 4801 must be a decimal number greater than 1000. The port numbers must be unique within the /etc/services file and the port number for interrupts (4801) is equal to the port number for the main connection (4800) plus one, for example, 4801=4800+1.

Refresh the inetd Daemon: As root user, synchronize the /etc/services file and the inetd daemon by executing the following commands:

```
# inetimp
# refresh -s inetd
```

Change the Database Manager Configuration: As the instance owner or as a sysadm, change the database manager configuration file by setting the service name value to the first port name in the /etc/services file. See "Define the TCP/IP Listen Ports for DataJoiner" on page 101.

```
$db2 update database manager configuration using svcename djmain
```

Set the DB2COMM Environment Variable to Enable TCP/IP: The environment variable DB2COMM specifies which communication protocol will be enabled when the database manager is started. The value for this variable is set in the .profile file of the instance owner. To enable TCP/IP, this variable must be either set to null, or have the value TCPIP as one of the allowed protocols. For our example, this variable was set to:

```
DB2COMM=APPC,TCPIP
export DB2COMM
```

But it could be also be set as follows:

```
DB2COMM=
export DB2COMM
```

If you make a change to the environment variable setting, as the DataJoiner instance owner, you must log off the system and login again, to make sure that DB2COMM is set to the proper value from the .profile file.

Start the DataJoiner Instance: Issue a db2start command for this instance. This starts two listen processes that wait for the remote-clients requests coming through the port numbers in the /etc/services file.

5.2.1.2 Configure DataJoiner to Access a Remote TCP/IP Server

For this example, we considered the DataJoiner instance djinst1 in the Severn system, to be the client. The steps to access the remote data source from this DataJoiner are the following:

1. Define the DataJoiner server host name and location to AIX.
2. Define the TCP/IP service ports to access the server.
3. Refresh the inetd daemon.
4. Catalog a node entry pointing to the location of the remote database.
5. Catalog the remote database in the system database directory.
6. Populate the System Catalog Tables for the DataJoiner data source.

Define the DataJoiner Server Host Name and Location to AIX: As root user, identify to your local system, if it is not already done, the host that has the DataJoiner or DB2/6000 data source to which you want to connect. You must update the local /etc/hosts file or a domain name server with the TCP/IP name

of the DataJoiner server. You can do this through smit or edit the file directly with vi.

Figure 77 shows how to define the DataJoiner server host with smit. Issue the smit command and select the following options from the menus:

Further Configuration
Name Resolution
Hosts Table (/etc/hosts)
Add a Host.

```
                                Add a Host Name

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

                                                      [Entry Fields]
* INTERNET ADDRESS (dotted decimal)                  [9.113.36.208]
* HOST NAME                                          [yellow.sanjose.ibm.com]
  ALIAS(ES) (if any - separated by blank space)      [yellow]
  COMMENT (if any - for the host entry)              []
```

Figure 77. Definition of a TCP/IP Host Using SMIT

Define the TCP/IP Service Ports to Access the Server: As root user, update the /etc/services file. Define the same two port numbers used on the DataJoiner or DB2/6000 data source (See "Define the TCP/IP Listen Ports for DataJoiner" on page 101). The names do not have to match between the client and the server.

```
djmn            4800/tcp    #yellow DataJoiner
djmni           4801/tcp
```

Refresh the inetd Daemon: As root user, synchronize the /etc/services file and the inetd daemon by executing the following commands:

```
# inetimp
# refresh -s inetd
```

Catalog a Node Entry Pointing to the Location of the Remote Data Source: As the instance owner or as a sysadm user, catalog a TCP/IP node using the following command:

```
$ db2 catalog tcpip node toserver remote host server port
```

where

toserver Node entry name for use in the system database directory. This can be any name no longer than eight characters.

host TCP/IP name or IP address of the DataJoiner server. See "Define the DataJoiner Server Host Name and Location to AIX" on page 102

port Name of the first port number in the local /etc/services file. See section "Define the TCP/IP Service Ports to Access the Server" on page 103

For example, we used the following command to define the TCP/IP node for the remote server on the Yellow system:

```
$db2 catalog tcpip node TODJYELL remote yellow server djmn
```

Catalog the Remote Database in the System Database Directory: This step is required only if you want to access this data source using the CONNECT TO SQL statement.

As the instance owner or as a sysadm user, catalog the remote database using the following command:

```
$ db2 catalog database dbname as alias at node toserver authentication auth
```

where

dbname Name of the database on the DataJoiner or DB2/600 data source.

alias Alternative name for the database being cataloged.

toserver Specifies the node where DataJoiner server resides. This must match the name you specify in the node directory.

See "Catalog a Node Entry Pointing to the Location of the Remote Data Source" on page 103

auth This must match the authentication type of the database in the remote data source. It can be server, dcs or client.

For our case the remote database is created with authentication set to server.

For example, we used the following command:

```
$ db2 catalog database dbinst1 as YELLOWDJ \
> at node TODJYELL authentication server
```

Populate the System Catalog Tables for the DataJoiner Data Source: For each database within the remote DataJoiner or DB2/6000 instance, you must have a row in the SYSIBM.SYSSERVERS table. If the authentication type of the target database is server or dcs, you must also populate the SYSIBM.SYSREMOTEUSERS table with the authorization ID of the users at the DataJoiner client system that need to access this data source, along with their corresponding remote authorization ID and password for the DataJoiner or DB2/6000 data source. If the authorization ID and password are the same in the

local and the remote systems, you do not need to add a row in the SYSIBM.SYSREMOTEUSERS table.

This is a list of the steps to follow:

1. Login as the DataJoiner instance owner.
2. Connect to the DataJoiner local database. For our environment we used the following command:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row in the SYSIBM.SYSSERVERS table for the remote DataJoiner or DB2/6000 data source. In the following examples we inserted a row for a database with authentication type server, and a row for another database in the remote DataJoiner instance, which has the authentication type client.

   ```
   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
   > VALUES('DJYELT','TODJYELL','dbinst1','DATAJOINER','1.0','jra','Y')"

   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
   > VALUES('DJYELT1','TODJYELL','sample','DATAJOINER','1.0','jra','N')"
   ```

   where

   SERVER Name that you want to assign to the DataJoiner data source. You may select any unique identifier of 1-8 characters.

   NODE Must match the node name that you assign to this data source location in the node directory. See "Catalog a Node Entry Pointing to the Location of the Remote Data Source" on page 103.

   DBNAME Name of the database on the DataJoiner data source.

   TYPE DATAJOINER for a DataJoiner data source, and DB2/6000 for a DB2/6000 V1 or DB2/6000 V2 data source.

   VERSION Version of the data source. Use 1.0 for a DataJoiner or DB2/6000 V1. Use 2.0 for DB2/6000 V2. In our example, we used 1.0 for DataJoiner and for DB2/6000 V1, and used 2.0 for DB2/6000 V2.

   PROTOCOL Must be jra. This field is case sensitive.

   PASSWORD This field is Y if the authentication for the target database is server or dcs, and is N if the . authentication is client

   In this example, we considered only the columns of the SYSBIM.SYSSERVERS table that cannot be set to null. These nonnullable columns represent the minimum information you need to connect to the DataJoiner or DB2/6000 data source.

4. Populate the SYSIBM.SYSREMOTEUSERS table:

   If the authentication type of the target database is server or dcs, you must insert a row for each of the users that need to access the remote data source. Issue the following command:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES(authid, server, remote_authid, remote_pw)"
```

where

authid	User authorization ID on the DataJoiner system
server	Name of the DataJoiner server as defined in SYSIBM.SYSSERVERS
remote_authid	User authorization ID for the remote DataJoiner server
remote_password	User password for the DataJoiner data source.

The following example shows the row we inserted in the SYSIBM.SYSREMOTEUSERS table for the DataJoiner instance owner:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES('djinst1','DJYELT','djinst1','DJINST1')"
```

You should now be able to create nicknames for the remote data source tables you need to access from DataJoiner. You can also issue the `set passthru servername` command to access the DataJoiner or DB2/6000 data source directly, where *servername* is the server name assigned to one of the databases of the DataJoiner data source in the SYSIBM.SYSSERVERS table: `djyelt` in our example.

5.2.2 Other DataJoiner Database in the Same DataJoiner Instance

If you have two DataJoiner databases within the same instance, and you want one of them to be a data source for the other, you must do the following:

1. Configure DataJoiner for TCP/IP clients:
   a. Define the TCP/IP listen ports for DataJoiner
   b. Refresh the inetd daemon.
   c. Change the database manager configuration.
   d. Set the DB2COMM environment variable to enable TCP/IP.
   e. Start the DataJoiner instance.
2. Configure DataJoiner to access a DataJoiner TCP/IP server.
   a. Catalog a node entry pointing to the location of the remote database.
   b. Populate the system catalog tables for the DataJoiner data source.
   c. Update the database manager configuration to allow two or more databases to be active concurrently.

To perform these steps, we assume that the minimum TCP/IP configuration is set; that is, you have a host name, a TCP address, and a domain name.

For our example, Figure 76 on page 101 shows the minimum TCP/IP configuration for the Severn system.

5.2.2.1 Configuring DataJoiner for TCP/IP Remote Clients

For the example on this section, we worked with two databases created under the same instance of DataJoiner in the Severn system. We configured this DataJoiner for TCP/IP clients.

1. Define the TCP/IP listen ports for DataJoiner
2. Refresh the inetd daemon.
3. Change the database manager configuration.
4. Set the DB2COMM environment variable to enable TCP/IP.
5. Start the DataJoiner instance.

Define the TCP/IP Listen Ports for DataJoiner: As the root user, add two entries to the /etc/services file. You can do this either with the `smit` utility, or edit the file directly with `vi`. We defined the following ports in the Severn system:

```
severndj        4802/tcp     # DataJoiner main connection port
severndji       4803/tcp     # DataJoiner interrupt port
```

where severndj and severndji are any names not already in the /etc/services file. The numbers for the ports 4802 and 4803 must be a decimal number greater than 1000. The port numbers must be unique within the /etc/services file and the port number for interrupts (4803) must equal to the port number for the main connection (4802) plus one, for example, 4803=4802+1.

Refresh the inetd Daemon: As the root user, synchronize the /etc/services file and the inetd daemon. Issue the following commands:

```
# inetimp
# refresh -s inetd
```

Change the Database Manager Configuration: As the DataJoiner instance owner or as a `sysadm` user, change the database manager configuration file by setting the service name value to the first port name in the /etc/services file.

```
$db2 update database manager configuration using svcename severndj
```

Set the DB2COMM Environment Variable to Enable TCP/IP: The environment variable DB2COMM specifies which communication protocol will be enabled when the database manager is started. The value for this variable is set in the .profile file of the instance owner. To enable TCP/IP, this variable must either be set to null, or be given the value of `TCPIP` as one of the allowed protocols. For our example, this variable was set as follows:

```
DB2COMM=APPC,TCPIP
export DB2COMM
```

But it could also be set as follows:

```
DB2COMM=
export DB2COMM
```

If you make a change to the environment variable setting, you must, as the DataJoiner instance owner, log off the system and login again, to make sure that DB2COMM is set with the proper value.

Start the DataJoiner Instance: Issue a db2start command for this instance. This starts two listen processes that wait for the remote-clients requests coming through the port numbers in the /etc/services file.

5.2.2.2 Configure DataJoiner to Access a DataJoiner TCP/IP Data Source

In this section, the DataJoiner instance that is in the Severn system is the server and also the client. The steps to access the other database in the same instance as a data source are these:

1. Catalog a node entry pointing to the location of the remote database.
2. Populate the system catalog tables for the DataJoiner data source.
3. Update the database manager configuration to allow two or more databases to be active concurrently.

Catalog a Node Entry Pointing to the Location of the DataJoiner Instance: As the instance owner or as a system administrator, catalog a TCP/IP node using the following command:

```
$ db2 catalog tcpip node toserver remote host server port
```

where

toserver Node entry name for use in the system database directory. You may select any unique identifier of one to eight characters.

host TCP/IP name or IP address of the local host.

port Name of the first port number in the local /etc/services file, as defined in “Define the TCP/IP Listen Ports for DataJoiner” on page 107.

For example, we used the following command

```
$ db2 catalog tcpip node TODJSEV remote severn server severndj
```

Populate the System Catalog Tables for the DataJoiner Data Source: You must have a row in the SYSIBM.SYSSERVERS table for the data source. You do not need to add rows to the SYSIBM.SYSREMOTEUSERS table because the authorization ID and the password of the users that need to access this data source are always the same, as both the client and the data source are in the same system.

The steps we followed are:

1. Login as DataJoiner instance owner.
2. Connect to the DataJoiner client database. For our example:

```
$ db2 connect to dbinst1
```

3. Insert a row into the SYSIBM.SYSSERVERS table for the data source.

   In the following example we inserted a row for a database with an authentication type of server.

```
$ db2 "INSERT INTO SYSIBM.SYSSERVERS \
> (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
> VALUES('DJTPCC','TODJSEV','tpcc','DATAJOINER','1.0','jra','Y')"
```

   where

   SERVER Name that you want to assign to the DataJoiner data source. Select any unique identifier of one to eight characters.

   NODE Must match the node name assigned to this data source location in the node directory. See "Catalog a Node Entry Pointing to the Location of the DataJoiner Instance" on page 108.

   DBNAME Name of the second database on the DataJoiner instance data source.

   TYPE DATAJOINER for a DataJoiner data source.

   VERSION Version of the data source. Version is 1.0 for our example because it is a DataJoiner data source.

   PROTOCOL Must be jra. This field is case sensitive.

   PASSWORD This field is Y if the authentication for the target database is server or dcs. It is N if the authentication is client.

Allow Two or More Databases to be Active Concurrently: As the instance owner or a sysadm user, update the database manager configuration with the following command:

```
$ db2 update database manager configuration using numdb n
```

where n is the number of databases that can be active concurrently. This number must be at least two, to establish the connection between two databases within the DataJoiner instance. For example:

```
$ db2 update database manager configuration using numdb 2
```

You should now be able to create nicknames for the data source tables you need to access from DataJoiner. You can also issue the set passthru *servername* command to access the DataJoiner data source directly, where *servername* is the name you assigned to the other database of the DataJoiner instance data source in the SYSIBM.SYSSERVERS table. For our example, the name of the server was DJTPCC.

5.2.3 Another DataJoiner Instance in the Same System

From the point of view of connectivity, connecting DataJoiner is the same whether to an instance of DataJoiner, or DB2/6000 V1 or DB2/6000 V2, located in the same system. In this section we show how to connect DataJoiner to another instance of DataJoiner in the same system, and we assume that the same definitions work for an instance of DB2/6000 V1 or DB2/6000 V2. However, we mention the differences where they apply.

The steps required are:

1. On the data source DataJoiner instance, configure DataJoiner for TCP/IP remote clients:
   a. Define the TCP/IP listen ports for DataJoiner.
   b. Refresh the inetd daemon.
   c. Change the database manager configuration.
   d. Set the DB2COMM environment variable to enable TCP/IP.
   e. Start the DataJoiner instance.
2. On the client DataJoiner instance, configure DataJoiner to access a TCP/IP DataJoiner data source.
   a. Catalog a node entry pointing to the location of the data source.
   b. Catalog the remote database in the system database directory.
   c. Populate the communications catalog tables for the DataJoiner data source.

To be able to perform these steps, we assume that the minimum TCP/IP configuration is set; that is, you have a host name, a TCP address, and a domain name.

For our example, Figure 76 on page 101 shows the minimum TCP/IP configuration for the Severn system.

5.2.3.1 Configuring DataJoiner or DB2/6000 for TCP/IP Remote Clients

For this example, we considered the `djinst2` DataJoiner instance in the Severn system to be the data source for the `djinst1` DataJoiner instance. The steps to configure djinst2 DataJoiner instance for TCP/IP clients are:

1. Define the TCP/IP listen ports for DataJoiner.
2. Refresh the inetd daemon.
3. Change the database manager configuration.
4. Set the DB2COMM variable to enable TCP/IP.
5. Start the DataJoiner instance.

Define the TCP/IP Listen Ports for DataJoiner: As root user, add two entries to the /etc/services file. You can either do this with the `smit` utility, or edit the file directly with `vi`. In the Severn system, we defined the following ports for the `djinst2` DataJoiner instance:

```
sevdj2          4804/tcp     # DataJoiner main connection port
sevdj2i         4805/tcp     # DataJoiner interrupt port
```

Where .sevdj2 and sevdj2i are any names not already in the /etc/services file. The numbers for the ports 4804 and 4805 must be a decimal number greater than 1000. The port numbers must be unique within the /etc/services file and the port number for interrupts (4805) is equal to the port number for the main connection (4804) plus one, for example, 4805=4804+1.

Refresh the inetd Daemon: As root user, synchronize the /etc/services file and the inetd daemon by executing the following commands:

```
# inetimp
# refresh -s inetd
```

Change the Database Manager Configuration: As the instance owner for the data source instance, or as a sysadm user, change the database manager configuration file by setting the service name value to the first port name in the /etc/services file. See “Define the TCP/IP Listen Ports for DataJoiner” on page 110. We used the following command:

```
$db2 update database manager configuration using svcename sevdj2
```

Set the DB2COMM Environment Variable to Enable TCP/IP: The environment variable DB2COMM specifies which communication protocol will be enabled when the database manager is started. The value for this variable is set in the .profile file of the instance owner of the data source. To enable TCP/IP, this variable must be set to null, or be given the value of TCPIP as one of the allowed protocols. For our example, we set this variable for the djinst2 as follows:

```
DB2COMM=
export DB2COMM
```

If you make a change to the environment variable setting, as the data source instance owner, you must log off the system and login again, to make sure that DB2COMM environment variable is set to the proper value.

Start the DataJoiner Instance: Issue a db2start command for this instance. This starts two listen processes that wait for the remote-client requests coming through the port numbers in the /etc/services file.

5.2.3.2 Configure DataJoiner to Access a Remote TCP/IP Data Source

For the example in this section, we considered the djinst1 DataJoiner instance in the Severn system to be the client. The steps to access the data source from this DataJoiner are:

1. Catalog a node entry pointing to the location of the remote data source.
2. Catalog the remote database in the system database directory.
3. Populate the communications catalog tables for the DataJoiner data source.

Catalog a Node Entry Pointing to the Location of the Remote Data Source: As the instance owner for the client, or as a system administrator (sysadm) user, catalog a TCP/IP node using the next command:

```
$ db2 catalog tcpip node toserver remote host server port
```

where

toserver Node entry name for use in the system database directory. Select any unique identifier of one to eight characters.

host TCP/IP name or IP address of the DataJoiner server. For our example, this is the local host name: severn.

port Name of the first port number in the local /etc/services file. See “Define the TCP/IP Listen Ports for DataJoiner” on page 110.

For example, we issued the following command from the instance owner of djinst1:

```
$db2 catalog tcpip node TODJ2SEV remote severn server sevdj2
```

Catalog the Remote Database in the System Database Directory: This step is required only if you want to access this data source using the CONNECT TO SQL statement.

As the instance owner for the client, or as a system administrator (sysadm) user, catalog the remote database using the following command:

```
$ db2 catalog database dbname as alias at node toserver authentication auth
```

where

dbname Name of the database on the DataJoiner server. Use any unique identifier of one to eight characters.

alias Alternative name for the database that is being cataloged. Use any unique identifier of one to eight characters.

toserver Specifies the node where DataJoiner server resides. This must match the name you specified in the node directory. See “Catalog a Node Entry Pointing to the Location of the Remote Data Source.”

auth Must match the authentication type of the database in the data source. It can be server, dcs or client.

For our case the data source database was created with the authentication type set to server.

For example, we used the following command:

```
$ db2 catalog database dbinst2 as DBDJ2SEV \
> at node TODJ2SEV authentication server
```

Populate the Communications Catalog Tables for the DataJoiner Data Source: For each database within the data source instance, you must have a row in the SYSIBM.SYSSERVERS table. You do not need to add rows to the SYSIBM.SYSREMOTEUSERS table because the authorization ID and the password of the users that need to access this data source are always the same, because both the client and the data source are in the same system.

These are the steps we followed:

1. Login as the DataJoiner instance owner.
2. Connect to the client DataJoiner local database. We used the following command:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row in the SYSIBM.SYSSERVERS table for the remote DataJoiner data source. In the next example we inserted a row for a database with an authentication type of server:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
   > VALUES('DJ2SEV','TODJ2SEV','dbinst2','DATAJOINER','1.0','jra','Y')"
   ```

   where

   SERVER Name that you want to assign to the DataJoiner data source. Use any unique identifier of one to eight characters.

   NODE Must match the node name that you assigned to this data source location in the node directory. See section "Catalog a Node Entry Pointing to the Location of the Remote Data Source" on page 112

   DBNAME Name of the database on the data source. The name of the database on the djinst2 was dbinst2

   TYPE 'DATAJOINER' for a DataJoiner data source. 'DB2/6000' for a DB2/6000 V1 or DB2/6000 V2 data source.

   VERSION Version of the data source. Use 1.0 for a DataJoiner or DB2/6000 V1 data source. Use 2.0 for a DB2/6000 V2 data source.

   PROTOCOL Must be jra. This field is case sensitive

PASSWORD This field is Y if the authentication for the target database is server or dcs, and is N if the authentication is client.

You should now be able to create nicknames for the data source tables you need to access from DataJoiner. You can also issue the set passthru *servername* command to access the DataJoiner data source directly, where *servername* is the name you assign to one of the databases of the data source in the SYSIBM.SYSSERVERS table. For our example, the server is DJ2SEV.

5.2.4 Configuring DB2/2 V2 as a Data Source

To add DB2/2 V2 as a data source, perform the following tasks:

1. Update the /etc/hosts file on the DataJoiner machine.
2. Update the /etc/services file on the DataJoiner machine.
3. Refresh the inetd daemon on the DataJoiner machine.
4. Update the \etc\services file on the DB2/2 V2 machine.
5. Update the DB2/2 V2 database configuration.
6. Catalog the DB2/2 V2 node on the DataJoiner machine.
7. Catalog the DB2/2 V2 database on the DataJoiner machine.
8. Update the DataJoiner System Catalog tables.
9. Create nicknames.

5.2.4.1 Update the DataJoiner /etc/hosts file

You need to tell DataJoiner how to find the DB2/2 V2 data source using TCP/IP. Do this by adding an entry in the DataJoiner /etc/hosts file for the data source. Login as root and use SMIT or an editor to update the /etc/hosts file with the definition for the DB2/2 V2 machine. If you are using SMIT, enter smit at the command line and from the System Management window, select the following options:

Communications Applications and Services
TCP/IP
Further Configuration
Name Resolution
Hosts Table
Add a Host.

Update the Add a Host Name window with the information for your DB2/2 V2 data source and press Enter. Figure 78 on page 115 shows our Add a Host Name window.

```
                                Add a Host Name

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

                                                      [Entry Fields]
* INTERNET ADDRESS (dotted decimal)                  [9.113.44.225]
* HOST NAME                                          [java.sanjose.ibm.com]
  ALIAS(ES) (if any - separated by blank space)      [java]
  COMMENT (if any - for the host entry)              []

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 78. Add a Host Name

INTERNET ADDRESS (dotted decimal)
This is the internet address of the DB2/2 V2 data source. This value can be found on the Network page of the TCP/IP Configuration icon on the OS/2 desktop.

HOST NAME This value is in the format of machine_name.domain_name. These values can be found on the Services page of the TCP/IP Configuration icon on the OS/2 desktop.

ALIAS(ES) Choose an alias for this machine. We used the machine name.

Alternatively, using an editor such as vi, add the following line to the /etc/hosts file:

/etc/hosts

```
INTERNET ADDRESS    MACHINE_NAME.DOMAIN_NAME  ALIAS
```

Our entry was as follows:

Our /etc/hosts file

```
9.113.44.225    java.sanjose.ibm.com java
```

5.2.4.2 Update the DataJoiner /etc/services File

You need two listening ports for your DB2/2 V2 data source: the main connection port and the interrupt port. To add these, login as root and use SMIT or an editor to update the /etc/services file with the definitions for the DB2/2 V2 machine. If you are using SMIT, enter `smit` at the command line and from the System Management window, take the following options:

Communications Applications and Services
TCP/IP
Further Configuration
Client Network Services
Services (/etc/services)
Add a Service.

Update the Add a Service window with the information for your main connection port for the DB2/2 V2 data source and press Enter. Figure 79 shows our Add a Service window for the main connection port.

```
                                Add a Service

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

                                                      [Entry Fields]
* Official Internet SERVICE Name                     [java]
* Transport PROTOCOL                                  tcp
* Socket PORT number                                 [4806]
  Unofficial Internet SERVICE NAMES                  []
  (separate names with blanks)

F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Reset       F6=Command        F7=Edit           F8=Image
F9=Shell          F10=Exit          Enter=Do
F9=Shell          F10=Exit          Enter=Do
```

Figure 79. Add a Service (main connection port)

Official Internet SERVICE Name — Choose any name that is unique in the /etc/services file.

Transport PROTOCOL — Use the value `tcp`.

Socket port NUMBER — This value must be an even number that is unique to the /etc/services file. It must match the value in the OS/2 \etc\services file configured in 5.2.4.4, “Update the DB2/2 V2 Services File” on page 118.

Now, from the Add a Service window, press PF3 to return to the Services window. Select Add a Service again and update the Add a Service window with the information for the DB2/2 V2 interrupt port this time and press Enter. Figure 80 on page 117 shows our Add a Service window for the interrupt port.

```
                                Add a Service

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

                                                      [Entry Fields]
* Official Internet SERVICE Name                     [javai]
* Transport PROTOCOL                                  tcp
* Socket PORT number                                 [4807]
  Unofficial Internet SERVICE NAMES                  []
  (separate names with blanks)

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
F9=Shell            F10=Exit            Enter=Do
```

Figure 80. Add a Service (interrupt port)

Official Internet SERVICE Name — Choose any name that is unique in the /etc/services file. The convention is to use the connection port name with an i on the end to represent interrupt. See the Official Internet SERVICE name entered in Figure 79 on page 116.

Transport PROTOCOL — Use the value tcp.

Socket port NUMBER — This value must be 1 plus the number used for the main connection port. See Figure 79 on page 116 for the value used there. This must also match the value in the OS/2 \etc\services file for the interrupt port which is configured in 5.2.4.4, “Update the DB2/2 V2 Services File” on page 118.

Alternatively, using an editor such as vi, add the following lines to the /etc/services file:

/etc/services

```
service_name    socket_port_number/tcp
service_namei   socket_port_number+1/tcp
```

The values we used are as follows:

Our /etc/services file

```
java            4806/tcp        DB2/2 V2 source on Java
javai           4807/tcp        DB2/2 V2 source on Java
```

5.2.4.3 Refresh the inetd Daemon

After you have made all your changes to the /etc/services file, you must refresh the inetd daemon. Login as root and issue the following commands:

```
inetimp
refresh -s inetd
```

This makes your new definitions available to TCP/IP. If you make more changes to /etc/services, you must run these commands again to make the latest changes available to TCP/IP.

5.2.4.4 Update the DB2/2 V2 Services File

This file usually resides in a directory called **x**:\tcpip\etc where **x**: represents the OS/2 drive where TCP/IP is installed. Using any OS/2 editor, add an entry for the DB2/2 V2 main connection port and the DB2/2 V2 interrupt port. These entries must correspond to the entries for this data source in the DataJoiner /etc/services file which were configured in 5.2.4.2, “Update the DataJoiner /etc/services File” on page 116. The port numbers and transport protocol *must* be the same in both files. The names can be different but it makes sense to have them the same.

The lines added to our \tcpip\etc\services file were as follows:

OS/2 services file

```
java                    4806/tcp
javai                   4807/tcp
```

To make these changes available to TCP/IP, restart TCP/IP on the data source. This can be done by issuing the following command from an OS/2 command prompt:

```
tcpstart
```

5.2.4.5 Update the DB2/2 V2 Database Configuration

To update the database configuration on OS/2, perform the following steps:

1. Open the IBM DATABASE 2 icon.
2. Open the Database Directory icon.
3. Select the database.
4. From the action bar, select **Selected**.
5. From the pull down menu, select **Configure**.

On the DB2 Configure window, select the **Protocols** tab.

Enter values for the workstation name and service name. The service name is the name defined in the OS/2 services file. This was configured in 5.2.4.4, “Update the DB2/2 V2 Services File.” The workstation name is the *machine_name* defined in 5.2.4.1, “Update the DataJoiner /etc/hosts file” on page 114. Our window is shown in Figure 81.

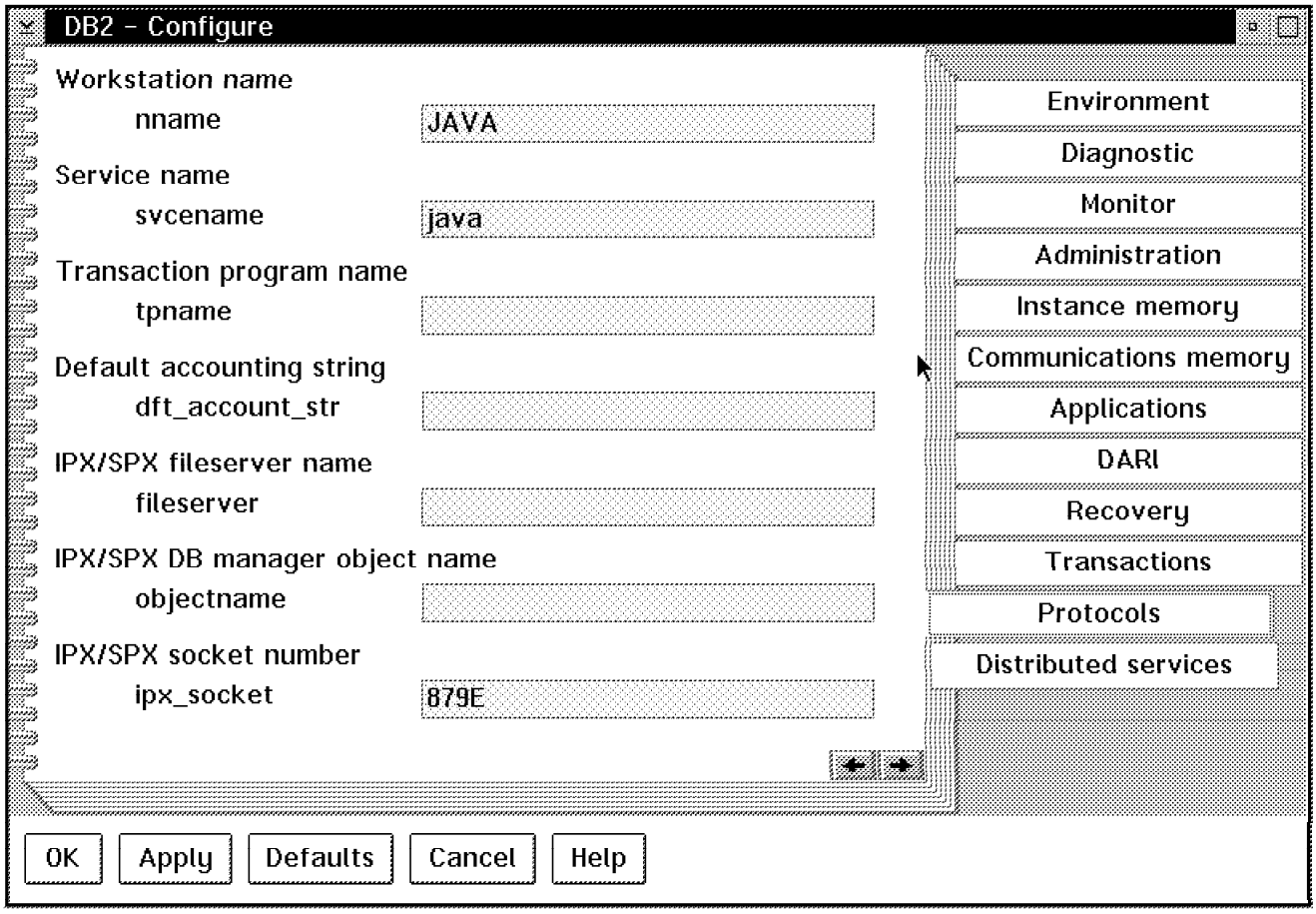


Figure 81. DB2 Configure window

5.2.4.6 Catalog a Node Entry on DataJoiner

Login as the instance owner or as a system administrator (sysadm) user and catalog the DB2/2 TCP/IP node using the following command:

```
$ db2 catalog tcpip node toserver remote host server port
```

Substitute the following:

toserver Choose a node name for use in the system database directory. This can be any name no longer than eight characters.

host TCP/IP name or IP address of the DataJoiner data source. See 5.2.4.1, “Update the DataJoiner /etc/hosts file” on page 114 and use the HOST NAME or ALIAS defined there.

port Name of the first port number in the local /etc/services file. See 5.2.4.2, “Update the DataJoiner /etc/services File” on page 116.

For example, we used the following command to define the TCP/IP node for the DB2/2 V2 data source:

```
$db2 catalog tcpip node TOJAVA remote JAVA server
java
```

5.2.4.7 Catalog the Remote Database in the System Database Directory

This step is required only if you want to access DB2/2 V2 using the CONNECT TO SQL statment.

As the instance owner or as a system administrator (sysadm) user, catalog the remote database using the following command:

```
$ db2 catalog database dbname as alias at node toserver authentication auth
```

Substitute the following:

dbname Name of the database on the DB2/2 V2 data source.

alias Alternative name for the database that is being cataloged.

toserver Specifies the node where the DB2/2 data source resides. This must match the name you specify in the node directory.

See 5.2.4.6, “Catalog a Node Entry on DataJoiner” on page 119.

auth This must match the authentication type of the database in the remote data source. It can be server, dcs or client.

For our case the remote database is created with authentication set to server.

For example, we used the following command:

```
$ db2 catalog database sample as DBJAVA \
> at node TOJAVA authentication server
```

5.2.4.8 Populate the System Catalog Tables for the DataJoiner Data Source

For each database in the DB2/2 V2 instance, you must have a row in the SYSIBM.SYSSERVERS table. If the authentication type of the target database is server or dcs, you must also populate the SYSIBM.SYSREMOTEUSERS table with the authorization ID of the DataJoiner users who need to access the data source. You must also include their corresponding remote authorization ID and password for the DB2/2 V2 data source. If the authorization ID and password are the same

in the local and the remote systems, you do not need to add a row to the SYSIBM.SYSREMOTEUSERS table.

Follow these steps:

1. Login as the DataJoiner instance owner.
2. Connect to the DataJoiner local database. For our environment we used the following command:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row in the SYSIBM.SYSSERVERS table for the DB2/2 data source. In the following example, we inserted a row for a database with an authentication type of server:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
   > VALUES('DJJAVA','TODJAVA','sample','DB2/2','2.0','jra','Y')"
   ```

   where

   SERVER Name that you want to assign to the DataJoiner data source. You may select any unique identifier of 1-8 characters.

   NODE Must match the node name that you assign to this data source location in the node directory. See 5.2.4.6, "Catalog a Node Entry on DataJoiner" on page 119.

   DBNAME Name of the database on the DB2/2 V2 data source.

   TYPE Use DB2/2 for a DB2/2 data source.

   VERSION Version of the data source. Use 2.0 for DB2/2 V2.

   PROTOCOL Must be jra. This field is case sensitive.

   PASSWORD This field is Y if the authentication for the target database is server or dcs, and is N if the authentication is client

   In this example, we considered only the columns of the SYSBIM.SYSSERVERS table that cannot be set to null. These nonnullable columns represent the minimum information you need to connect to the DB2/2 V2 data source.

4. Populate the SYSIBM.SYSREMOTEUSERS table.

   If the authentication type of the target database is server or dcs, you must insert a row for each of the users that need to access the remote data source. Issue the following command:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
   > VALUES(authid, server, remote_authid, remote_pw)"
   ```

   where

   authid User authorization ID on the DataJoiner system

   server Name of the DataJoiner server as defined in SYSIBM.SYSSERVERS

remote_authid User authorization ID on the DB2/2 data source

remote_password User password on the DB2/2 data source.

The following example shows the row we inserted in the SYSIBM.SYSREMOTEUSERS table for the DataJoiner instance owner:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES('djinst1','DJJAVA','userid','password')"
```

You should now be able to create nicknames for the remote data source tables you need to access from DataJoiner. Refer to 7.3.4, "Create Nicknames for Remote Tables and Views" on page 165 for information on creating nicknames. You can also issue the set passthru *servername* command to access the DB2/2 V2 data source directly. The *servername* is the server name you assigned to the database in the DB2/2 V2 data source in the SYSIBM.SYSSERVERS table. Our example uses DJJAVA

5.2.5 DataJoiner or DB2/6000 V1 in a Remote Machine Using APPC

TCP/IP is the recommended protocol to access a remote DataJoiner or DB2/6000 V1 data source. However, we are including this section as an alternative form of establishing the connection.

To access a remote DataJoiner or DB2/6000 data source through APPC, the steps are these:

1. APPC configuration for the data source:
   a. Configure the initial node setup.
   b. Configure the local LU 6.2.
   c. Configure an LU 6.2 mode profile.
   d. Configure a TPN profile.
   e. Verify the configuration profiles.
   f. Update the database manager configuration.
2. APPC configuration for the client:
   a. Configure the initial node setup.
   b. Configure a token-ring link-station profile.
   c. Configure the local LU 6.2.
   d. Configure a LU 6.2 side-information profile.
   e. Configure a LU 6.2 mode profile.
   f. Configure a partner LU 6.2 location profile.
   g. Verify the configuration profiles.
   h. Start the SNA link station.
   i. Catalog a node in the node directory.
   j. Catalog a database in the system database directory.
   k. Populate the system catalog tables.

In the examples for this section, we considered the DataJoiner in the Severn system as the client, and the DataJoiner in the Yellow system as the data source. See Figure 75 on page 99.

5.2.5.1 APPC Configuration for the Data Source

The steps to configure the APPC data source are these:

1. Configure the initial node setup.
2. Configure the local LU 6.2.
3. Configure an LU 6.2 mode profile.
4. Configure a TPN profile.
5. Verify the configuration profiles.
6. Update the database manager configuration.

Configure the Initial Node Setup: Login as root and issue the `smit sna` command to go into the SNA Server/6000 main menu.

Select the following options from the menus to go to the initial setup window:

Configure SNA profiles
Initial Node Setup.

For the type of data link control we want to configure, we selected `token_ring`, giving us the window shown in Figure 82

```
                              Initial Node Setup

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

                                                    [Entry Fields]
  Control Point name token_ring              +     [SCA2109]
  Control Point type                               appn_end_node
  Local network name                               [USIBMSC]
  XID node ID                                      [071a2109]

  Optional link station information:

  Link station type                                token_ring
  Link station name                                []
* Calling link station?                            yes
  Link address                                     []
```

Figure 82. Initial Node Setup Window on Yellow

Define the following fields:

Control point name Any name you want, but it can be the PU name.

Control point type Must be `appn_end_node`.

Local network name The name for your local network. For our example, this is USIBMSC.

XID node ID Must be 071a plus a four-character identifier of your system. For our example, this is 2109.

Leave the other fields with the default values, and press Enter to save the profile.

Configure the Local LU 6.2: Exit from the initial node setup window and return to the Configure SNA profiles menu. Select the following options to go to the LU 6.2 local LU window:

Advanced configuration
Sessions
LU 6.2
LU 6.2 Local LU.

You can choose to add or change a profile depending on whether you have already defined this profile. Figure 83 shows the way we defined our local LU 6.2.

```
                        Add LU 6.2 Local LU Profile

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

                                                    [Entry Fields]
  Profile name                                     [SCA2109I]
  Local LU name                                    [SCA2109I]
  Local LU alias                                   [SCA2109I]
  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                                         []
```

Figure 83. Local LU Profile on Yellow

You must define the following fields:

Profile name Any name you want, but it can be the LU name.

Local LU name LU name that you want to assign to the system.

Local LU alias Alternative name for the local LU. We used the same local LU name identifier.

Local lu is dependent?
Must be set to no.

Leave the other fields with the default values, and press Enter to save the profile.

Configure a LU 6.2 Mode Profile: Exit from the Add LU 6.2 local LU Profile window and return to the LU 6.2 menu. Select the following option to go to the LU 6.2 mode menu:

LU 6.2 Mode

You can choose to add or change a profile depending on whether you have already defined this profile. Figure 84 shows the way we defined our local LU.

```
                        Add LU 6.2 Mode Profile

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

                                                      [Entry Fields]
  Profile name                                       [IBMRDB]
  Mode name                                          [IBMRDB]
  Maximum number of sessions (1-5000)                [16]
  Minimum contention winners (0-5000)                [8]
  Minimum contention losers (0-5000)                 [8]
  Auto activate limit (0-500)                        [0]
  Upper bound for adaptive receive pacing window     [16]
  Receive pacing window (0-63)                       [7]
  Maximum RU size (128,...,32768: multiples of 32)   [1024]
  Minimum RU size (128,...,32768: multiples of 32)   [256]
  Class of Service (COS) name                        [#CONNECT]

  Comments                                           []
```

Figure 84. IBMRDB Mode Profile on Yellow

Define the following fields:

Profile name Can be any name, but you can use IBMRDB for a relational database connection.

Mode name Name you assign to the mode. You can use the same identifier as in the profile name.

Leave the other fields with the default values, and press Enter to save the profile.

Configure a Transaction Program Name Profile: Exit from Add the LU 6.2 Mode Profile window and return to the LU 6.2 menu, and select the following option to go to the LU 6.2 transaction program name menu:

LU 6.2 Transaction Program Name (TPN).

You must define two transaction profiles. Figure 85 on page 126 and Figure 86 on page 127 show the profiles for our example.

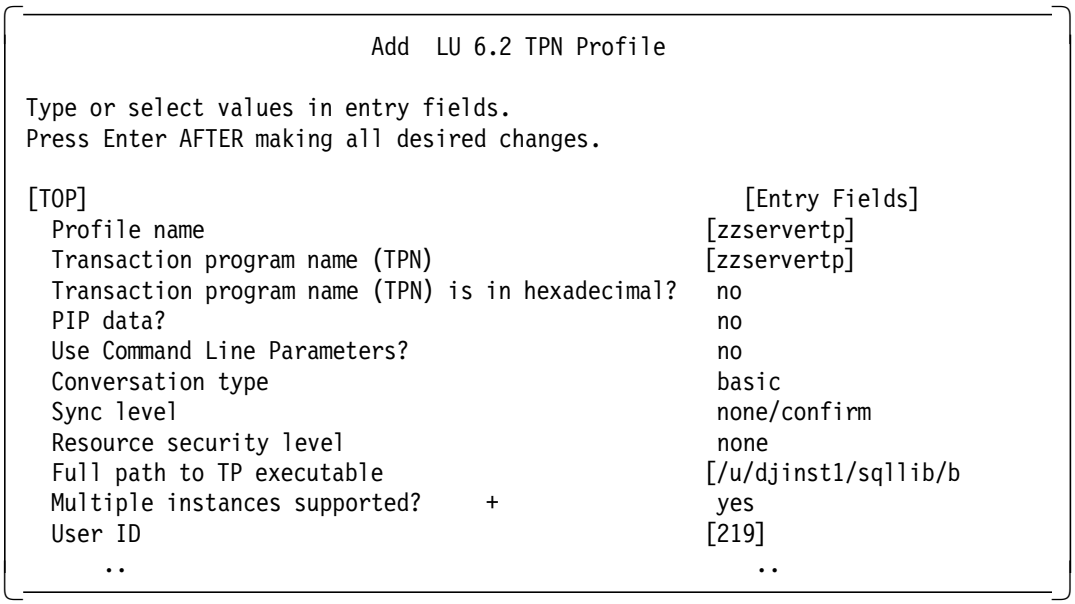


Figure 85. Transaction Program Name for APPC Clients on Yellow

Define the following fields:

Profile name Can be any name.

Transaction program name
Can be any name.

Conversation type Set it to basic

Full path to TP executable
Type /u/djinst1/sqllib/bin/db2acntp where /u/djinst1 is the directory of the instance owner.

Multiple instances supported?
Set to yes.

User ID User ID number of the instance owner.

Leave the other fields with the default values, and press Enter to save the profile.

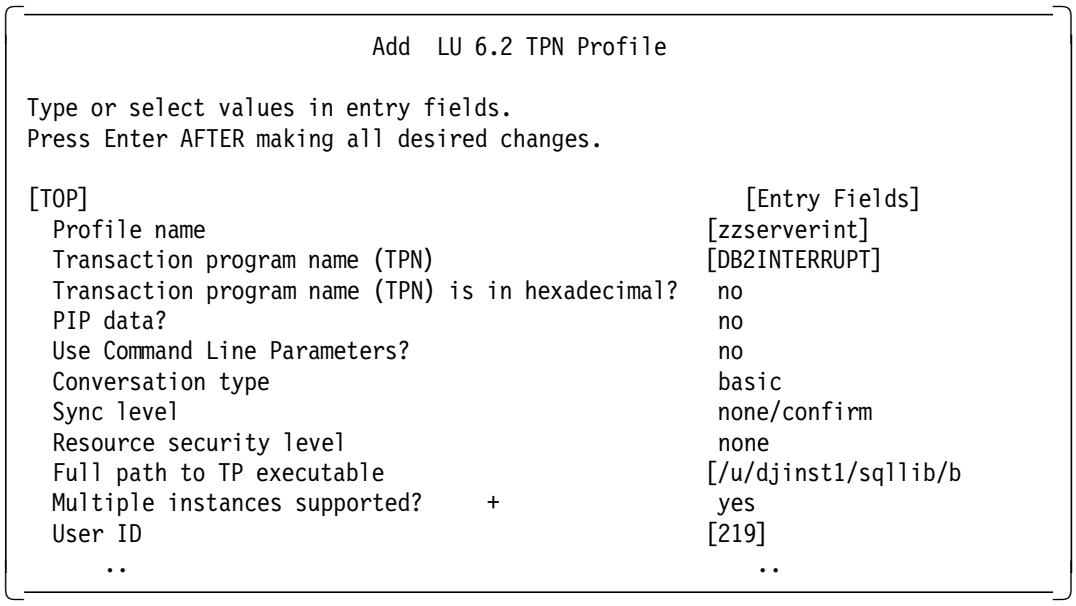


Figure 86. Interrupts Transaction Program Name Profile on Yellow

Define the following fields:

Profile name Can be any name.

Transaction program name
Must be DB2INTERRUPT.

Conversation type Set to basic.

Full path to TP executable
Type /u/djinst1/sqllib/bin/db2alttp where /u/djinst1 is the directory of the instance owner.

Multiple instances supported?
Set to yes.

User ID User ID number of the instance owner. To get this value enter the id at the command prompt of the instance owner.

Leave the other fields with the default values, and press Enter to save the profile.

Verify the Configuration Profiles: Exit from the LU 6.2 TPN window and press the F3 key repeatedly until you get to the Advanced configuration menu, and select:

Verify configuration profiles.

Set the update action field to dynamic_update with the tab key, and press Enter to verify the profiles. If the verification is successful then exit SMIT by pressing the F10 key. If verification is not successful, return to the configuration menus to fix the problem.

Update the Database Manager Configuration: Login as the instance owner and run the following command:

```
$ db2 update database manager configuration using tpname tpn
```

Where *tpn* is the name you assigned to the TPN profile in “Configure a Transaction Program Name Profile” on page 125. For our example, the command is:

```
$ db2 update database manager configuration using tpname zzservertp
```

Stop the DataJoiner instance with the db2stop command. Ensure that the DB2COMM environment variable, which is in the instance owner .profile file, is set to allow the APPC protocol to be enabled. Start the DataJoiner instance again. This variable can be defined as shown in the following example to enable both the APPC and the TCP/IP protocols:

```
DB2COMM=
export DB2COMM
```

5.2.5.2 APPC Configuration for the Client

For our example, we used the DataJoiner that was in the Severn system as the client. The steps to configure the DataJoiner APPC client are described in the following sections and are as follows:

1. Configure the initial node setup.
2. Configure a token ring link-station profile.
3. Configure the local LU 6.2.
4. Configure an LU 6.2 side-information profile.
5. Configure an LU 6.2 mode profile.
6. Configure a Partner LU 6.2 location profile.
7. Verify the configuration profiles.
8. Start the SNA link station.
9. Catalog a node in the node directory.
10. Catalog a database in the system database directory.
11. Populate the system catalog tables.

5.2.5.3 Configure the Initial Node Setup

As in “Configure the Initial Node Setup” on page 123, you have to configure the initial node setup for the client system. Figure 87 on page 129 shows the definition for the Severn system.

```
                              Initial Node Setup

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

                                                    [Entry Fields]
  Control Point name token_ring              +     [SCA2085]
  Control Point type                               appn_end_node
  Local network name                               [USIBMSC]
  XID node ID                                      [071a2085]

  Optional link station information:

  Link station type                                token_ring
  Link station name                                []
* Calling link station?                            yes
  Link address                                     []
```

Figure 87. Interrupts Transaction Program Name Profile on Severn

Configure a Token Ring Link Station Profile: Login as root and issue the `smit sna` command. Select the following options through the menus to go to the Add Token Ring Link Station Profile window, as shown in Figure 88 on page 130:

Configure SNA Profiles
Advanced Configuration
Links
Token Ring
Token Ring Link Station
Add a Profile.

```
                    Add Token Ring Link Station Profile

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

[TOP]                                                  [Entry Fields]
  Profile name                                        [trlnkyel]
  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
     ....
  Adjacent Node Address Parameters
     Access routing                                    link_address
     If link_address,
        Remote link address                           [400052047184]
        Remote SAP address (02-fa)                    [04]

  Adjacent Node Identification Parameters
     Verify adjacent node?                             no
     Network ID of adjacent node                      [USIBMSC]
     CP name of adjacent node[USIBMSC]                 ]
     XID node ID of adjacent node (LEN node only)      *]
     Node type of adjacent node                        learn
     ....
```

Figure 88. Token Ring Link Station Profile on Severn

Define the following fields:

Profile name Any name you want.

SNA DLC profile Press F4 on this field and select tok0.0001.

Remote link address Token ring address of the server.

Network ID of the adjacent node
Name of the network for the server.

Leave the other fields with the default values, and press Enter to save the profile.

Configure the Local LU 6.2: As in "Configure the Local LU 6.2" on page 124, you must configure the LU 6.2 local LU for the client system. Figure 89 on page 131 shows the definition for the Severn system.

```
                    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                               SCA2085I
  New profile name                                   []
  Local LU name                                      [SCA2085I]
  Local LU alias                                     [SCA2085I]
  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                                           []
```

Figure 89. Interrupts Transaction Program Name Profile on Severn

Configure a LU 6.2 Side Information Profile: Exit from the Change/Show LU 6.2 local LU profile window, and return to the LU 6.2 menu. Select the following options:

LU 6.2 Side Information
Add a Profile.

Figure 90 shows the way we define the side information profile for our example.

```
                  Add LU 6.2 Side Information Profile

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

                                                   [Entry Fields]
* Profile name                                    [DJYELTPN]
  Local LU or Control Point alias                 [SCA2085I]
  Provide only one of the following:
     Partner LU alias                             []
     Fully qualified partner LU name              [USIBMSC.SCA2109I]
  Mode name                                       [IBMRDB]
  Remote transaction program name (RTPN)          [zzservertp]
  RTPN in hexadecimal?                             no

  Comments                                        [DJ in yellow]
```

Figure 90. Side Information Profile on Severn to Access Yellow DataJoiner

Define the following fields:

Profile name Any name you want.

Local LU or Control Point alias
The name of LU 6.2 local LU profile. See “Configure the Local LU 6.2” on page 130.

Fully qualified partner LU name
The name of the network and the LU name of the partner system. In our case the partner system is Yellow.

This information is in the Initial node setup window of Yellow, and in the LU 6.2 local LU profile. See “Configure the Initial Node Setup” on page 123 and “Configure the Local LU 6.2” on page 124.

Mode name Must match the LU 6.2 mode profiles names defined in both the client and the data source. See “Configure a LU 6.2 Mode Profile” on page 125 and 5.2.5.4, “Configure a LU 6.2 Mode Profile”

Remote transaction program name (RTPN)
Must match the first transaction program name profile defined in the data source system. For our example, this value was “zzservertp.” See “Configure a Transaction Program Name Profile” on page 125.

Leave the other fields with the default values, and press Enter to save the profile.

5.2.5.4 Configure a LU 6.2 Mode Profile

As in “Configure a LU 6.2 Mode Profile” on page 125, you must configure the LU 6.2 mode for the client system. Figure 91 shows the definition for our example.

```
                    Change/Show LU 6.2 Mode Profile

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

                                                      [Entry Fields]
  Current profile name                                IBMRDBM
  New profile name                                   []
  Mode name                                          [IBMRDBM]
  Maximum number of sessions (1-5000)                [10]
  Minimum contention winners (0-5000)                [5]
  Minimum contention losers (0-5000)                 [5]
  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)   [2816]
  Minimum RU size (128,...,32768: multiples of 32)   [1024]
  Class of Service (COS) name                        [#CONNECT]

  Comments                                           []
```

Figure 91. Mode Profile on Severn

Configure a Partner LU 6.2 Location Profile: Exit from the Change/Show LU 6.2 Mode Profile Window, and return to the LU 6.2 menu. Select the following options:

Partner LU 6.2 Location
Add a profile

Figure 92 shows the Partner LU 6.2 Location Profile for our case.

```
                    Add Partner LU 6.2 Location Profile

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

                                                      [Entry Fields]
* Profile name                                       [DJYELPLU]
  Fully qualified partner LU name                    [USIBMSC.SCA2109I]
  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                                   [SCA2085I]
     Link Station Profile name                       [trlnkyel]

  Comments                                           [DJ in yellow]
```

Figure 92. Partner LU 6.2 Location Profile to Access Yellow DataJoiner

Define the following fields:

Profile name Any name you want.

Fully qualified partner LU name
The name of the network and the LU name of the partner system. In our case this system was Yellow.

This information is in the Initial Node Setup window of the Yellow system, and in the LU 6.2 local LU profile. See "Configure the Initial Node Setup" on page 123 and "Configure the Local LU 6.2" on page 124. See also "Configure the Local LU 6.2" on page 130.

Partner LU location method
Set it to `link_station`.

Local LU name Must match the LU 6.2 local LU profile name defined in the client.

See "Configure the Local LU 6.2" on page 130

Link Station Profile name
Must match the link station profile name defined in the client to access the data source. See "Configure a Token Ring Link Station Profile" on page 129.

Leave the other fields with the default values, and press Enter to save the profile.

Verify the Configuration Profiles: Exit from the Add Partner LU 6.2 Location Profile window and press the F3 key repeatedly until you get to the Advanced configuration menu, then select:

Verify configuration profiles

Set the update action field to dynamic_update with the tab key and press Enter to verify the profiles. If the verification is not successful, return to the configuration menus to fix the problem.

Start the SNA Link Station: Exit from the Verify Configuration Profile window and press F3 repeatedly until you get to the SNA Server/6000 menu. Select the following options:

Manage SNA resources
Start SNA Resources
Start an SNA link station.

In the Start an SNA Link Station window, press F4, and select from the list the name of the link station profile you defined in "Configure a Token Ring Link Station Profile" on page 129. Press Enter to start the link station and leave SMIT.

Catalog a Node in the Node Directory: Log in as the instance owner or as a system administrator (sysadm) user, and catalog a CPIC node using the next command:

```
$ db2 catalog cpic node toserver remote tpn \
> security sec
```

where

toserver Node entry name for use in the system database directory.

tpn Name of the side information profile defined to access the server. See "Configure a LU 6.2 Side Information Profile" on page 131.

sec If the authentication type of the target database is set to server or DCS, set it to program, and if the authentication type of the target database is set to client, set it to same.

For example, we used the following command:

```
$db2 catalog cpic node TODJYEL remote DJYELTPN \
security program
```

Catalog a Database in the System Database Directory: This step is required only if you want to access this data source using the CONNECT TO SQL statment.

As the instance owner or as a system administrator (sysadm), catalog the remote database using the following command:

```
$ db2 catalog database dbname as alias \
at node toserver authentication auth
```

where:

dbname Name of the database on the data source.

alias Alternative name for the database being cataloged.

toserver Specifies the node where the data source resides. This must match the name you specified in the node directory. See "Catalog a Node in the Node Directory" on page 134.

auth Must match the authentication type of the database in the data source. This can be server, dcs or client.

For our example the remote database was created with authentication type server.

For example, we used the following command:

```
$ db2 catalog database DBINST1 as DBDJYEL \
> at node TODJYEL authentication server
```

Populate the System Catalog Tables: For each database within the data source instance, you must have a row in the SYSIBM.SYSSERVERS table. If the authentication type of the target database is server or dcs, you must also populate the SYSIBM.SYSREMOTEUSERS table with the authorization ID of the DataJoiner users who need to access this data source, along with their corresponding remote authorization ID and password for the DataJoiner server. However, if the authorization ID and password are the same for both the client and the data source, you need not insert the row in the SYSIBM.SYSREMOTEUSERS table.

These are the steps we followed:

1. Login as the DataJoiner instance owner.
2. Connect to the client DataJoiner local database. We used the following command:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row into the SYSIBM.SYSSERVERS table for the remote DataJoiner data source. We use the following SQL statment:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
   > VALUES('DJYELA','TODJYEL','dbinst1','DATAJOINER','1.0','jra','Y')"
   ```

   where

SERVER	Name you want to assign to the data source.
NODE	Must match the node name that you assigned to this data source location in the node directory. See "Catalog a Node in the Node Directory" on page 134.
DBNAME	Name of the database on the data source.
TYPE	DATAJOINER for a DataJoiner data source, and DB2/6000 for a DB2/6000 V1.
VERSION	Version of the DataJoiner data source. In our case 1.0, but it is the same for DB2/6000 V1.
PROTOCOL	Must be jra. This field is case sensitive.
PASSWORD	This field is Y if the authentication for the target database is server or dcs, and is N if the authentication is client.

In our example, we considered only the columns of the SYSBIM.SYSSERVERS table that cannot be set to null. These nonnullable columns represent the minimum information you need to connect to the data source.

4. Populate the SYSIBM.SYSREMOTEUSERS table:

   If the authentication type of the target database is server or dcs, you must insert a row for each of the users who need to access the data source. You must issue the following command:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES(authid, server, remote_authid, remote_pw)"
```

   where

authid	User authorization ID on the client system
server	Name of the data source server as defined in the SYSIBM.SYSSERVERS table
remote_authid	User authorization ID for the data source system
remote_password	User password for the data source system.

   The following example shows the row we inserted in the SYSIBM.SYSREMOTEUSERS table for the DataJoiner instance owner:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES('djinst1','DJYELA','djinst1','DJINST1')"
```

You should now be able to create nicknames for the data source tables you need to access from DataJoiner. You can also issue the set passthru *servername* command to access the data source directly, where *servername* is the name you assigned to one of the databases of the data source in the SYSIBM.SYSSERVERS table. For our example the server name was DJYELA.

5.3 Configuring DataJoiner to Access Non-IBM Data Sources

This section describes the steps required to connect DataJoiner to a non-IBM relational data source using TCP/IP.

Figure 93 shows the data source environment in which we worked, using *Oracle* and *Sybase* as our data sources.

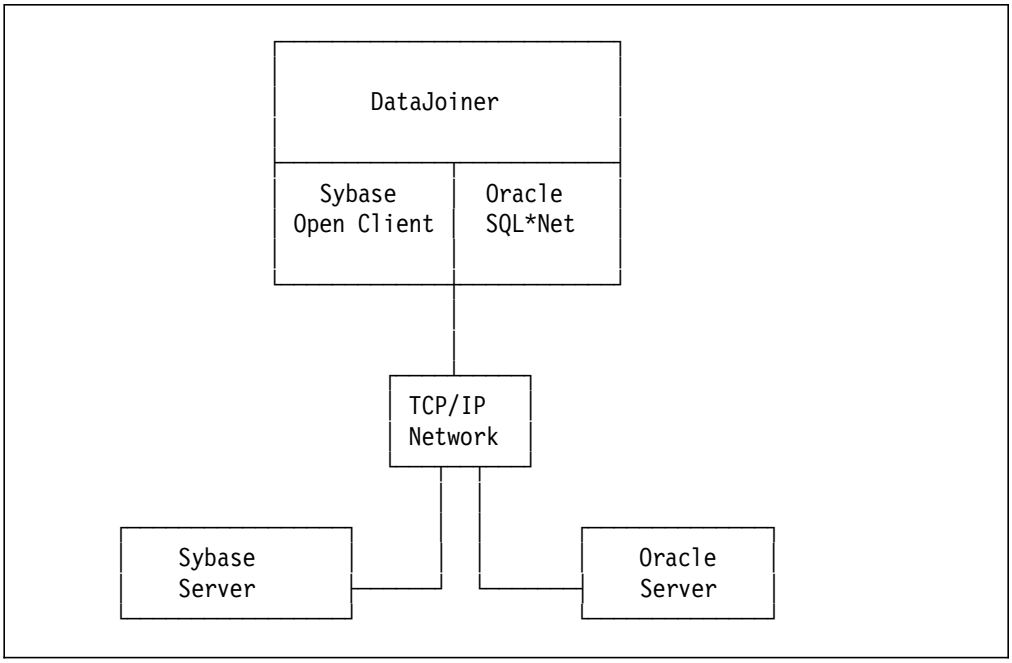


Figure 93. Non-IBM Relational Data Sources Environment

To follow the steps presented in this section, you should be aware that the non-IBM software must be installed in accordance with the supplier specifications. It is not the purpose of this document to give detailed information about how to install and configure non-IBM software, so we assume that the installation of the software is complete. However, we list the software required to establish the connection between DataJoiner and the non-IBM data sources, and the basic steps required to configure the DataJoiner client.

5.3.1 Configuring Sybase Data Access Module

In summary, the steps to connect DataJoiner to a Sybase data source are:

1. Get information from the Sybase server database administrator.
2. Install the Sybase software on the DataJoiner client and Sybase data source.
3. Link-edit the Sybase Open Client libraries to the DataJoiner Sybase data access module.

4. Create an interfaces file for the DataJoiner instance owner.
5. Populate the DataJoiner System Catalog Tables for the Sybase data source.

5.3.1.1 Get Information from the Sybase Server Database Administrator

You must get the following information from the Sybase database administrator to establish the connection:

1. User ID and password to access the data source.
2. TCP/IP address and network name for the data source.
3. An interfaces file with the required definitions to access the data source.
4. Name of the database in the data source.

5.3.1.2 Install the Sybase Software on the DataJoiner Client and Sybase Server

The Sybase data access module is included with the DataJoiner product. To enable the module to communicate with the Sybase data source, you must install, on the Sybase *server*, the *Sybase cataloged stored procedures*. On the DataJoiner system that is the *client*, you must install the *Sybase Open Client* libraries.

Refer to the appropriate Sybase documentation for information on how to install and configure this software.

5.3.1.3 Link Edit the Sybase Open Client Libraries to the DataJoiner Sybase Data Access Module

To access a Sybase data source, you must statically link DataJoiner to the Open Client libraries of the Sybase data source. Do the following:

1. Login as root.
2. With `vi`, edit the `djxlink` shell script installed with DataJoiner. It is in the `/usr/lpp/djx_01_01_0000/lib` directory.
3. Uncomment the statements pertaining to Sybase. Figure 94 on page 139 shows the lines we uncommented in the djxlink file for Sybase.
4. Set the SYBASE environment variable to point to the directory where Open Client is installed. Figure 94 on page 139 shows the value of the SYBASE environment variable for our environment.
5. Run the script with the command:

   ```
   /usr/lpp/djx_01_01_0000/lib/djxlink
   ```

```
# ------------------------------------------------------------------
# Optional - Sybase:
#
#            You only need to set these variables if you plan to
#            use Sybase as a data source and have the appropriate
#            Sybase licenses.  (If you already have Sybase set in
#            your environment, you do not need to set it here.)
# ------------------------------------------------------------------
Sybase=/home/sybase/OCLIENT
djxLibSybase="-lsybdb -L$Sybase/lib"

echo "Sybase=$Sybase"
echo "djxLibSybase=$djxLibSybase"
```

Figure 94. Statements in djxlink File for Sybase

The djxlink file also contains the statements to statically link the Oracle and DRDA data sources. Therefore, whenever you run this script, the statements of any data source that you want to access from DataJoiner must be uncommented.

5.3.1.4 Create an Interfaces File for the DataJoiner Instance Owner

There should be an interfaces file in the Open Client installation directory. The file points to the Sybase servers that you want to access. In our case the directory was /home/sybase/OCLIENT. Copy the interfaces file to the sqllib directory of the DataJoiner instance owner.

Figure 95 shows the interfaces file for our example. We issued the following command to copy the file to the DataJoiner instance owner:

```
$ cp /home/sybase/OCLIENT/interfaces /home/djinst1/sqllib
```

```
#
SybaseMVP 0 0
        query tcp ibm-ether mvpdb1 2510
        master tcp ibm-ether mvpdb1 2510
```

Figure 95. Sample Sybase Interfaces File

The interfaces file contains the information required to access the Sybase data source. Within the parameters that are defined in this file, there is one for the name of the node in which the data source resides. For our example, the name of the node is SybaseMVP. This name is used to define the server in the SYSIBM.SYSSERVERS table of DataJoiner.

5.3.1.5 Populate the DataJoiner System Catalog Tables for Sybase Data Source

For each database within each Open Server defined in the interfaces file, you must have a row in the SYSIBM.SYSSERVERS table. You must also populate the SYSIBM.SYSREMOTEUSERS table with the authorization ID of the DataJoiner users who need to access Sybase with their corresponding remote authorization ID and password for the Sybase server.

These are the steps we followed:

1. Login as DataJoiner instance owner.
2. Connect to the DataJoiner database. This is the local database. In our example the command is:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row in the SYSIBM.SYSSERVERS table for the Sybase data source, for example:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSSERVERS \
   > (SERVER, NODE, DBNAME, TYPE, VERSION, PROTOCOL, PASSWORD) \
   > VALUES('sybase1','SybaseMVP','test_db','SYBASE','4.5','openclient','Y')"
   ```

   where

   SERVER Name that you want to assign to the Sybase server.

   NODE Name must match the definition in the interfaces file.

   DBNAME Name of the database on the Sybase server.

   TYPE SYBASE

   VERSION Version of the Sybase software. In our case this is 4.5.

   PROTOCOL Must be openclient. This field is case sensitive.

   PASSWORD Y. The user ID and password are verified at the Sybase server.

   In this example, we considered only those columns of the SYSBIM.SYSSERVERS table that cannot be set to null. The nonnullable columns represent the minimum information you need to connect to Sybase.

4. Populate the SYSIBM.SYSREMOTEUSERS table:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
   > VALUES(AUTHID, SERVER, REMOTE_AUTHID, REMOTE_PW)"
   ```

   where

   AUTHID User authorization ID on the DataJoiner system.

   SERVER Name of the Sybase server as defined in the SYSIBM.SYSSERVERS table.

   REMOTE_AUTHID User authorization ID on the Sybase server.

   REMOTE_PASSWORD User password on the Sybase server.

   We inserted the following row in the SYSIBM.SYSREMOTEUSERS table for the DataJoiner instance owner:

   ```
   $ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
   > VALUES('djinst1','sybase1','iwdj01','iwdj01')"
   ```

You should now be able to create nicknames for the Sybase tables you want to access from DataJoiner. You can also issue the set passthru sybase1 command to access the Sybase server directly, where *sybase1* is the name you assign to the Sybase server in the SYSIBM.SYSSERVERS table.

5.3.2 Configuring Oracle Data Access Module

In summary, the steps to connect DataJoiner to an Oracle data source are these:

1. Get information from the Oracle server database administrator.
2. Install the Oracle software on the client and server.
3. Link-edit the SQL*Net libraries to the DataJoiner Oracle call interface data access module.
4. Set the ORACLE_HOME environment variable for the DataJoiner instance.
5. Configure the tnsnames.ora file.
6. Populate the DataJoiner System Catalog Tables for the Oracle data source.

5.3.2.1 Get Information from the Oracle Server Database Administrator

You must get the following from the Oracle administrator to establish the connection:

1. User ID and password to access the data source.
2. TCP/IP address and network name for the data source.
3. Service port number of the Oracle listener to include it in the `/etc/services` file.
4. A tnsnames.ora file with the required definitions to access the data source.

5.3.2.2 Install the Oracle Software on the Client and Server

The Oracle data access module is included with the DataJoiner product. To enable the module to communicate with the Oracle data source through TCP/IP, you must install the following Oracle software in the DataJoiner system:

- SQL*Net
- TCP/IP Protocol Adapter.

For our environment, we installed the following versions of the products:

- SQL*Net V2 2.0.15.0.0
- TCP/IP Protocol Adapter (V2) 2.0.15.0.0.

Refer to the appropriate Oracle documentation for information on how to install and configure this software.

5.3.2.3 Link-Edit the SQL*Net Libraries to the DataJoiner Oracle Call Interface Data Access Module

To access an Oracle data source, you must statically link DataJoiner to the libraries of the Oracle data source. Do the following steps:

1. Login as root.
2. With `vi` edit the `djxlink` shell script installed with DataJoiner. This script is in the `/usr/lpp/djx_01_01_0000/lib` directory.
3. Uncomment the statements pertaining to Oracle. Figure 96 shows the Oracle part of the djxlink file modified for our environment.
4. Set the ORACLE_HOME environment variable to point to the directory where SQL*Net is installed. Figure 96 shows the ORACLE_HOME environment variable for our environment.
5. Run the script with the command:

   ```
   /usr/lpp/djx_01_01_0000/lib/djxlink
   ```

```
# ------------------------------------------------------------------
# Optional - Oracle
#
#            You only need to set these variables if you plan to
#            use Oracle as a data source and have the appropriate
#            Oracle licenses.  (If you already have Oracle_HOME set
#            in your environment, you do not need to set it here.)
# ------------------------------------------------------------------
Oracle_HOME=/home/Oracle/SQL*Net
djxLibOracle="$Oracle_HOME/lib/osntab.o -lnlsrtl -locic -lcore -lora
-lcv6 -lsqlnet -lnetwork -L$Oracle_HOME/lib"

echo "Oracle_HOME=$Oracle_HOME"
echo "djxLibOracle=$djxLibOracle"
```

Figure 96. Statements in djxlink File for Oracle

This file also contains the statements to statically link the Sybase and DRDA data sources. Therefore, whenever you run this script, the statements of any data source that you want to access from DataJoiner must have all comments removed.

5.3.2.4 Set the ORACLE_HOME Environment Variable for the DataJoiner Instance

You must set the ORACLE_HOME environment variable before starting the DataJoiner instance. This variable must point to the directory where SQL*Net is installed. In our case we set this variable in the .profile file of the DataJoiner instance owner with the following statement:

```
export Oracle_HOME=/home/Oracle/SQL*Net ;
```

If you change the ORACLE_HOME environment variable, you must stop and start the DataJoiner instance for the variable to take effect. DataJoiner uses the variable from the instance owner only and not from any other user.

5.3.2.5 Configure the tnsnames.ora File

You must update the SQL*Net tnsname.ora file for each Oracle server you want to access from DataJoiner. This file is usually located in the /etc directory. If the TNS_ADMIN environment variable is set in the environment of the DataJoiner instance, then it is used to locate the tnsnames.ora file.

Figure 97 shows the /etc/tnsnames.ora file that we used for our environment. We did not set the TNS_ADMIN environment variable.

```
###########
mvpo =
  (DESCRIPTION =
    (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = TCP)
          (HOST = mvpdb0.stl.ibm.com)
          (PORT = 1521)
        )
    )
    (CONNECT_DATA =
      (SID = mvpo)
    )
  )
```

Figure 97. Oracle tnsnames.ora File for our Environment

The tnsnames.ora file contains the information required to access the Oracle data source. Within the parameters that are defined in this file there is one for the name of the node in which the data source resides. For our example, the name of the node is mvpo. This name is used to define the server in the SYSIBM.SYSSERVERS table of DataJoiner.

5.3.2.6 Populate the DataJoiner System Catalog Tables for the Oracle Data Source

You must have a row in the SYSIBM.SYSSERVERS table for each instance that you define in the tnsnames.ora file, You also must populate the SYSIBM.SYSREMOTEUSERS table with the authorization ID of the DataJoiner users who want to access Oracle with their corresponding remote authorization ID and password for the Oracle server.

Here are the steps we followed:

1. Login as Datajoiner instance owner.
2. Connect to the local DataJoiner database. This is the local database. In our case:

   ```
   $ db2 connect to dbinst1
   ```

3. Insert a row in the SYSIBM.SYSSERVERS table for the Oracle data source:

```
$ db2 "INSERT INTO SYSIBM.SYSSERVERS \
> (SERVER,NODE,DBNAME,TYPE,VERSION,PROTOCOL,PASSWORD) \
> VALUES('ORACLE1','mvpo','','ORACLE','7.0','sqlnet','Y')"
```

where

SERVER Name that you want to assign to the Oracle server.

NODE Name must match the definition in the tnsnames.ora file.

DBNAME Blank because Oracle has only one database per instance.

TYPE ORACLE

VERSION Version of the Oracle server software. In our case 7.0.

PROTOCOL Must be sqlnet. This field is case sensitive

PASSWORD Y. The user ID and password are verified at the Oracle server.

In this example, we considered only the nonnullable columns of the SYSBIM.SYSSERVERS table. The nonnullable columns represent the minimum information you need to connect to Oracle.

4. Populate the SYSIBM.SYSREMOTEUSERS table:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
> VALUES(AUTHID, SERVER, REMOTE_AUTHID, REMOTE_PW)"
```

where

AUTHID User authorization ID on the DataJoiner system.

SERVER Name of the Oracle server as defined in the SYSIBM.SYSSERVERS table.

REMOTE_AUTHID User authorization ID for the Oracle server.

REMOTE_PASSWORD User password for the Oracle server.

We inserted the following row in the SYSIBM.SYSREMOTEUSERS table for the DataJoiner instance owner:

```
$ db2 "INSERT INTO SYSIBM.SYSREMOTEUSERS \
VALUES('djinst1','ORACLE1','iwdj01','iwdj01')"
```

You should now be able to create nicknames for the Oracle tables you want to access from DataJoiner. You can also issue the set passthru oracle1 command to access the Oracle server directly, where *oracle1* is the name you assign to the Oracle server in the SYSIBM.SYSSERVERS table.

Chapter 6. Security

This chapter describes how remote data source security, password validation, APPC security, and authentication types work together to control access to remote data sources. Scenarios for the various security options are also provided.

6.1 Overview

To gain access to a remote data source using DataJoiner, it is necessary to pass a number of security checkpoints along the way. Where these checkpoints are placed is determined mostly by the structure of your overall network. It is also determined by the existing security imposed upon you, plus security constraints that may be added as a result of new components.

A user must first gain system access to DataJoiner and subsequently gain access to the remote data source(s). Once system access to DataJoiner is obtained, a user gains data access by:

- Authorization to access the nicknames (See 7.1, “Enabling Users for Access to Local and Remote Data” on page 161) by being granted the privilege in DataJoiner.
- Authorization to access the actual data by being granted the privilege at the remote data source.

6.2 Implementing Security

Generally you are at the mercy of the security requirements imposed by the remote data sources. It is unlikely that you will be in a position to implement security changes to an environment that has been in place for some time. Therefore, we consider implementing security by conforming to the requirements of the remote data source, working outward through DataJoiner to the client; that is, from right to left in Figure 98 on page 146.

Let us first examine the general considerations of authentication.

6.2.1 Concept of Authentication

Authentication is the process of validating a *user ID* and password when an attempt is made to connect to a database. Because DataJoiner operates in a multiplatform environment, this validation can be done

- At the client workstation *only*
- At the client workstation *and* the remote data source(s)
- At the DataJoiner workstation *only*
- At DataJoiner *and* the remote data source(s)
- At the remote data source(s).

It is also possible to avoid authentication entirely. How and where authentication takes place depends upon the following factors, shown in Figure 98 on page 146, reading from right to left:

1 Remote data source security

© Copyright IBM Corp. 1995

2 Remote data source password validation specified for the remote data source at the time of cataloging

3 APPC security, the security specified for the APPC connections between the client and DataJoiner as well as between DataJoiner and the remote data source

4 DataJoiner authentication type, the authentication type specified for the DataJoiner database at the time of cataloging. The authentication type for the client must be compatible with the authentication type on DataJoiner.

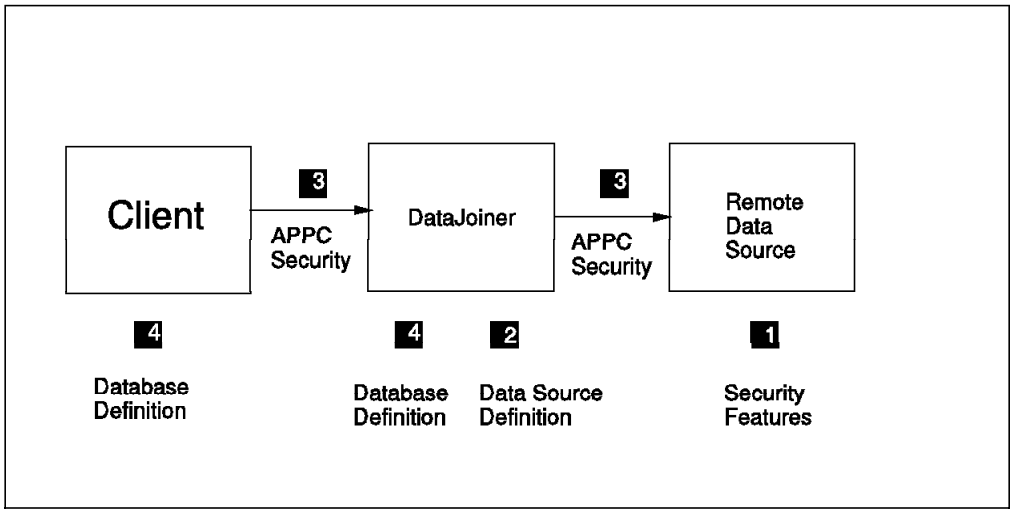


Figure 98. Authentication Factors

6.2.2 Remote Data Source Security

The first thing to consider is whether or not your remote data source is capable of running in "trusted client" mode, meaning that the data source owners have enough confidence in the security implementation of clients that no further validation is needed. See **1** in Figure 98. A remote data source running in trusted-client mode does not require that clients send a password, but it does require the authorization ID so that it can determine which privileges to allow the client. See Table 4 for a summary of trusted-client capabilities of DBMSs which are supported by DataJoiner.

Table 4 (Page 1 of 2). Trusted Client Implementation

DBMS	Trusted Client?	Implementation
DataJoiner	Yes	Create DB
DB2/2	No	
DB2/6000	Yes	Create DB
DB2 PE	Yes	Create DB
DB2 for HP-UX	Yes	Create DB
DB2 for Solaris	Yes	Create DB
DB2/MVS	Yes	SYSLUNAMES
SQL/DS (VM)	Yes	AVS USER TABLE
SQL/DS (VSE)	Yes	CICS/APPC Option
DB2/400	Yes	DEVICE DESCRIPTION

Table 4 (Page 2 of 2). Trusted Client Implementation		
DBMS	**Trusted Client?**	**Implementation**
Oracle	Yes	OPS$ login
Sybase	Yes	“trusted” option for user
via EDA/SQL	Yes	
via CrossAccess	Yes	

6.2.3 Remote Password Option

The PASSWORD column of the SYSIBM.SYSSERVERS table determines whether or not a password will be forwarded to the remote data source. See **2** in Figure 98 on page 146. You must ensure that valid parameter combinations are selected in order to obtain the desired result. See 6.3, “Security Scenarios” on page 149 for further guidance in this area.

If your remote data source runs in trusted-client mode, it is not necessary to forward a password. The password should then be coded as N in the SYSSERVERS table.

If the remote data source does *not* run in trusted client mode, then the authorization ID and password will be validated at the remote data source. It is then necessary to forward a password from DataJoiner to the remote data source. To do so, the value of PASSWORD should be Y in the SYSSERVERS table.

6.2.3.1 Which Password is Used?

When SYSIBM.SYSSERVERS.PASSWORD is set to Y, DataJoiner sends a password to the remote data source. The question is, which password should it send?

DataJoiner takes the authorization ID that was used to connect to the DataJoiner database and searches SYSIBM.SYSREMOTEUSERS for a remote authorization ID and password to send to the remote data source. If none is found, then DataJoiner will pass along the authorization ID and password used to connect to DataJoiner. If a remote authorization ID is found in the table, then it is sent to the remote data source. If a remote password is also specified in the table, then it is also sent to the remote data source. Otherwise, the DataJoiner password is used.

If the value of SYSIBM.SYSSERVERS.PASSWORD is set to N, no password will be sought from SYSIBM.SYSREMOTEUSERS and no password will be sent. Only the authorization ID is relevant.

Note: Although the password is stored in an encrypted form in the DataJoiner database, it flows *in the clear* to the remote data source.

See 5.1.4, “Update the System Catalog Tables” on page 97 for information on how to insert and update data to SYSIBM.SYSSERVERS and SYSIBM.SYSREMOTEUSERS.

6.2.4 APPC Security

APPC can be used as the communications protocol between the clients and DataJoiner, between DataJoiner and IBM relational databases, or both. See 3 in Figure 98 on page 146.

When APPC is used, conversation security is specified when the CPI-C node is cataloged in the node directory (by way of the CATALOG CPIC NODE command).

Security=PROGRAM
Specifies that both the user ID and password are included in the APPC allocation request which is sent to the server.

Security=NONE Specifies that neither the user ID nor the password is included in the APPC allocation request sent to the server

Note: Security=NONE is not permitted between DataJoiner and remote data sources.

Security=SAME Specifies that a user ID is included on the APPC allocation request sent to the server, together with an indicator that the user ID has already been verified. If you choose this option, then the VTAM APPL macro specification for the DB2 or SQL/DS host must have the parameter SECACPT set to ALREADYV.

6.2.5 Authentication Parameters

When a database is cataloged on either a DataJoiner workstation or a client workstation, you must specify where authentication is to take place. See 4 in Figure 98 on page 146. The options are:

Authentication=Client
The user ID and password are validated on the client workstation. Users are expected to be authenticated at the location from which they first sign on. For local DataJoiner users, the user ID may be taken from the user that issued the `login` command. Passwords do not flow to DataJoiner, but may flow to the remote data source, depending upon the value of SYSIBM.SYSSERVERS.PASSWORD.

Authentication=Server
The user ID and password are validated on the DataJoiner workstation. A valid user ID/password combination must be defined to the AIX security system. See 6.2.6, “User IDs at DataJoiner Server” on page 149 and 6.2.7, “Passwords at DataJoiner Server” on page 149. Passwords flow to DataJoiner and may flow to the remote data source, depending upon the value of SYSIBM.SYSSERVERS.PASSWORD.

Note: You cannot specify authentication type for client workstations running under Extended Services for OS/2 2.0 and DB2/2, so DataJoiner treats requests coming from these clients as if the authentication type were set to SERVER.

Authentication=DCS The user ID and password are validated at the remote data source only. Therefore no user ID need be defined to AIX on the DataJoiner server. User ID and password flow to

DataJoiner and onward to the remote data source. In order to send a password on to the remote data source, it is necessary to set the value of SYSIBM.SYSSERVERS.PASSWORD to Y. A value of N will result in no password being sent to the remote data source, thus creating an error condition when DataJoiner attempts to authenticate.

6.2.6 User IDs at DataJoiner Server

DataJoiner can support a wide range of clients and data sources, each with its own method of implementing security. It is therefore necessary for DataJoiner to be able to work with a method common to all these methodologies in order to be able to deal with all clients and remote sources concurrently.

To simplify rules for user and group IDs, DataJoiner requires that they all be defined in lower case within the AIX security facilities. This requirement arises for two reasons:

- When user names and group names are used within DataJoiner, they are always rolled to upper case. DataJoiner is *not* case sensitive.
- When DataJoiner makes use of the AIX security facilities, however, the user names and group names are rolled to lower case.

The user ID can be passed to DataJoiner in either upper or lower case; DataJoiner will manipulate the user ID case as required.

6.2.7 Passwords at DataJoiner Server

DataJoiner does not manipulate passwords as it does user IDs. It works with the password in whatever form it is received, whether it be upper or lower case. The AIX security facility *is* sensitive to the case of a password. Therefore it is very important to send the password to DataJoiner in the proper case.

Because OS/2 user profile management (UPM) converts all user IDs and passwords to uppercase, it can send only upper-case passwords to DataJoiner. If password standards are to apply equally to *all* clients, then *all* clients must send upper-case passwords. Therefore, if all clients are sending upper-case passwords, the password on the AIX security system must be defined in upper case.

User ID/Password Summary

DataJoiner user IDs *must* be defined in lower case. In addition, if you plan to have a network that includes OS/2 clients, passwords *must* be defined in upper case, as this is a requirement for User Profile Management (UPM) of OS/2.

6.3 Security Scenarios

With a combination of the AUTHENTICATION parameter on the CATALOG DATABASE command,the SECURITY parameter on the CATALOG NODE command, and the value in the PASSWORD column of SYSSERVERS, six security scenarios will be addressed in order to show the different possibilities for

implementation. They consist of three types of clients connecting to two types of remote data sources:

- Connecting to an IBM Server
  - Local DataJoiner clients
  - Remote APPC clients
  - Remote TCP/IP clients
- Connecting to a non-IBM server
  - Local DataJoiner clients
  - Remote APPC clients
  - Remote TCP/IP clients.

6.3.1 Security for Local Clients Connecting to an IBM Server

The following options are available for controlling local user security:

- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on *both* the DataJoiner workstation *and* the remote data source(s).
- Users are authenticated on the remote data source *only*.

6.3.1.1 Authentication at DataJoiner Workstation Only

Because the client is running on the same machine as the server, Authentication = *CLIENT* and Authentication = *SERVER* have the same meaning. That is, authentication is to take place on the DataJoiner machine. (Authentication = *DCS* would mean that authentication is to take place at the remote data source.)

Therefore when the authentication at DataJoiner is set to either *CLIENT* or *SERVER*, the authorization ID and password are validated by the AIX security system at the DataJoiner workstation when you connect to DataJoiner. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid` for that user/server pair and forwards that `authid` (if found) or the local `authid` to the remote data source.

When set up in this manner, the value for PASSWORD in SYSSERVERS must be set to `N` indicating that no password is to be sent to the remote data source because the remote data source does not do password validation. The APPC security between DataJoiner and the remote data source must be set to `SAME` indicating that the security has already been checked, and the remote data source must be running in trusted-client mode.

6.3.1.2 Authentication at DataJoiner and Remote Data Sources

In order to achieve authentication at both DataJoiner and a remote data source, Authentication must be set to either *CLIENT* or *SERVER*. The user ID and password will be authenticated at DataJoiner when you attempt to connect. The value of PASSWORD in SYSSERVERS must be set to `Y`, causing a password to flow to the remote data source. DataJoiner accesses SYSREMOTEUSERS to find the remote `authid/password` for that user/server pair. If it finds them, it forwards them to the remote data source. If they are not found, then DataJoiner sends the local `authid` and `password`. The APPC security must be set to `PGM`, indicating that security has not been checked and that user ID and password are forwarded to the remote data source. The remote data source may not be running in trusted-client mode.

6.3.1.3 Authentication at the Remote Data Source Only

In order to achieve authentication at the remote data source, set Authentication to DCS. The user ID and password are then not authenticated at DataJoiner. The value of PASSWORD in SYSSERVERS must be set to Y, causing a password to flow to the remote data source. DataJoiner accesses SYSREMOTEUSERS to find the remote authid/password for that user/server pair, and if they are found, it forwards them to the remote data source. If they are not found, then DataJoiner sends local authid and password. The APPC security must be set to PGM indicating that security has not been checked and that user ID and password will be forwarded to the remote data source. The remote data source may not be running in trusted-client mode.

6.3.1.4 Summary—Local DataJoiner Clients Connecting to IBM Server

Figure 99 summarizes the parameter combinations for specifying security for local DataJoiner clients through to an IBM remote data source.

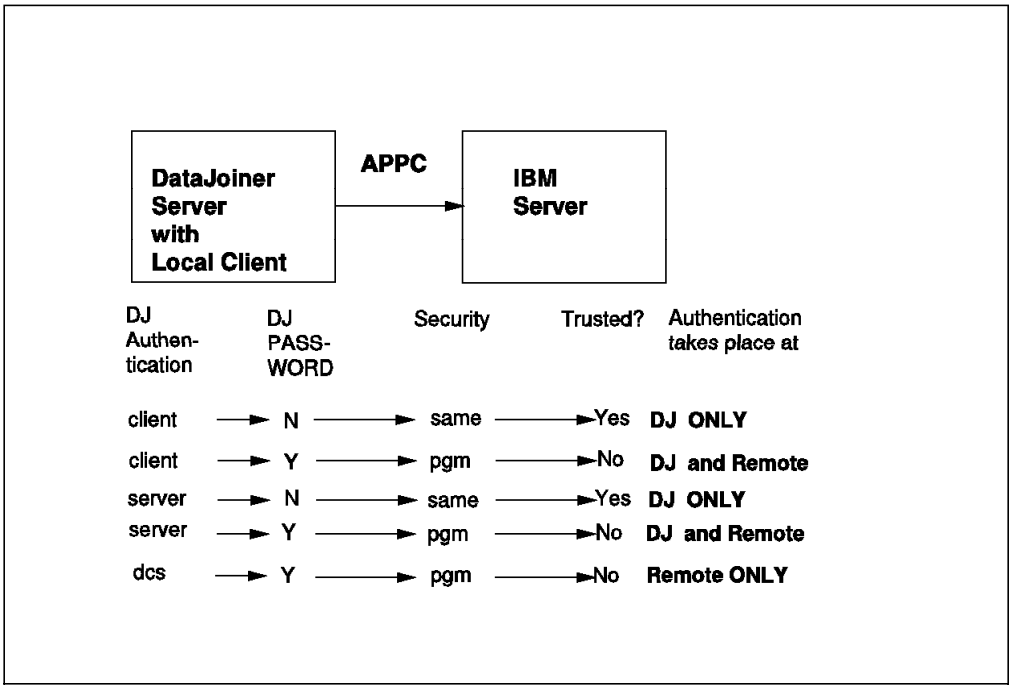


Figure 99. Security for Local DataJoiner Clients to an IBM Server

6.3.2 Security for Remote APPC Clients Connecting to an IBM Server

The following options are available for controlling remote user security:

- Users are authenticated on the Client workstation *only*.
- Users are authenticated on the Client workstation *and* the remote data source(s).
- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on BOTH the DataJoiner workstation *and* the remote data source(s).
- Users are authenticated on the remote data source *only*.

Because there is an extra APPC layer between the client and DataJoiner, you can specify a security classification at that APPC level as well. However, this layer of security is redundant and difficult to maintain. Therefore SECURITY = NONE is recommended at the client. SECURITY = NONE is not permitted on the connection to the remote data source.

6.3.2.1 Authentication at Client Workstation Only

When the authentication at the client is set to CLIENT, the authid and password are validated at the client when you log in. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote authid for that user/server pair and forwards that authid (if found) or the local authid to the remote data source.

When set up in this manner, the value for PASSWORD in SYSSERVERS must be set to N indicating that no password is to be sent to the remote data source because the remote data source does not validate passwords. The APPC security between DataJoiner and the remote data source must be set to SAME indicating that security has already been checked. The remote data source must be running in trusted-client mode.

6.3.2.2 Authentication at Client and Remote Data Sources

The client will be configured exactly the same as for authentication at DataJoiner only. (See 6.3.2.1, “Authentication at Client Workstation Only.”) However, slight differences apply to the DataJoiner workstation. Because we also wish to authenticate the remote data source, PASSWORD in SYSSERVERS must be set to Y, indicating that a password will be sent to the remote data source. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote authid and password for that user/server pair and forwards that authid/password (if found) or the local authid/password to the remote data source.

In this case APPC security between DataJoiner and the remote data source must be set to PGM, indicating that both user ID and password will be sent. The remote data source will not be running in trusted-client mode.

6.3.2.3 Authentication at DataJoiner workstation Only

From the point of view of the client, authentication will be performed somewhere down the line, either at DataJoiner or at the remote data source. Therefore, authentication at the client can be set to either SERVER or DCS. Authentication at DataJoiner must be set to SERVER, and PASSWORD must be set to N in SYSSERVERS. APPC security between DataJoiner and the remote data source must be set to SAME, indicating that security has already been verified. The remote data source must be running in trusted-client mode.

When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote authid for that user/server pair and forwards that authid (if found) or the local authid to the remote data source.

6.3.2.4 Authentication at DataJoiner and Remote Data Sources

The client will be configured exactly the same as for authentication at DataJoiner only. (See 6.3.2.3, “Authentication at DataJoiner workstation Only” on page 152.) However, slight differences apply to the DataJoiner workstation. Because we also wish to authenticate at the remote data source, PASSWORD in SYSSERVERS must be set to `Y`, indicating that a password will be sent to the remote data source. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid` and `password` for that user/server pair and forwards that `authid/password` (if found) or the local `authid/password` to the remote data source.

In this case APPC security between DataJoiner and the remote data source must be set to `PGM`, indicating that both user ID and password will be sent. The remote data source will not be running in trusted-client mode.

6.3.2.5 Authentication at Remote Data Source Only

From the point of view of the client, authentication will be performed somewhere down the line, either at DataJoiner or at the remote data source. Therefore authentication at the client can be set to either `SERVER` or `DCS`. Authentication at DataJoiner must be set to `DCS`, and PASSWORD must be set to `Y` in SYSSERVERS. APPC security between DataJoiner and the remote data source must be set to `PGM`, indicating that a user ID and password will be sent with the APPC allocation request. The remote data source must not be running in trusted client mode.

When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid/password` for that user/server pair and forwards that `authid/password` (if found) or the local `authid/password` to the remote data source.

6.3.2.6 Summary, Remote APPC Clients Connecting to IBM Server

Figure 100 on page 154 summarizes the combinations for specifying security from remote APPC clients through to an IBM remote data source.

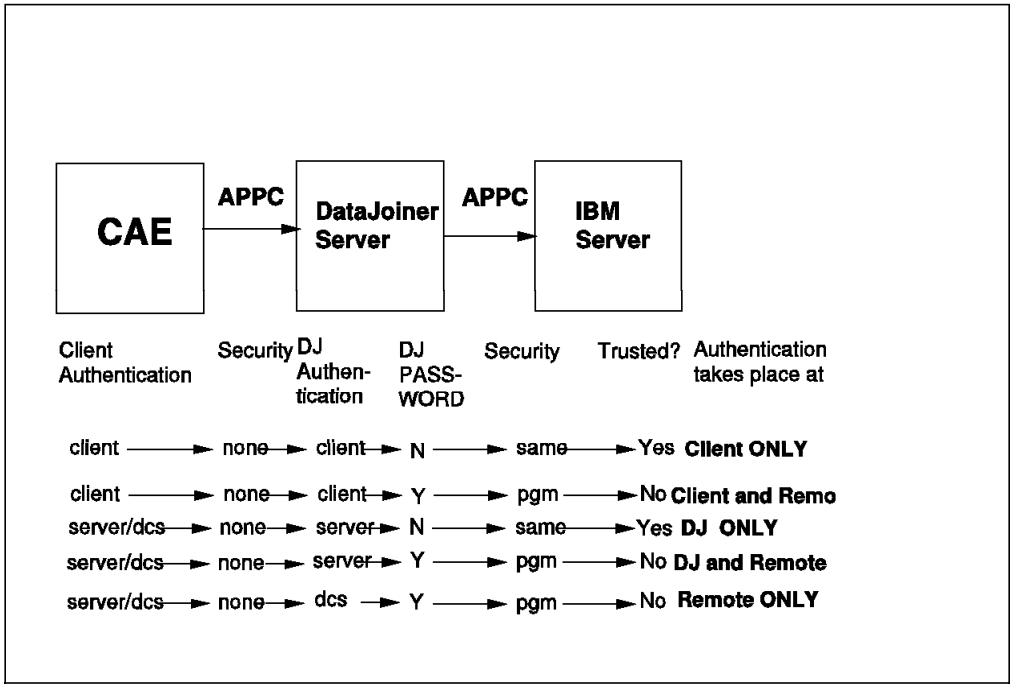


Figure 100. Security for Remote APPC Clients Connecting to an IBM Server

6.3.3 Security for Remote TCP/IP Clients Connecting to an IBM Server

The options for controlling remote user security of TCP/IP clients are the same as in 6.3.2, “Security for Remote APPC Clients Connecting to an IBM Server” on page 151. That is:

- Users are authenticated on the client workstation *only*.
- Users are authenticated on *both* the client workstation *and* the remote data sources.
- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on *both* the DataJoiner workstation AND the remote data source(s).
- Users are authenticated on the remote data source *only*.

The implementation is exactly the same as 6.3.2, “Security for Remote APPC Clients Connecting to an IBM Server” on page 151 except that no security level is specified between the client and DataJoiner. TCP/IP does not have any security option as APPC does.

6.3.3.1 Summary, Remote TCP/IP Clients Connecting to IBM Server

Figure 101 on page 155 summarizes the combinations for specifying security from remote TCP/IP clients through to an IBM remote data source.

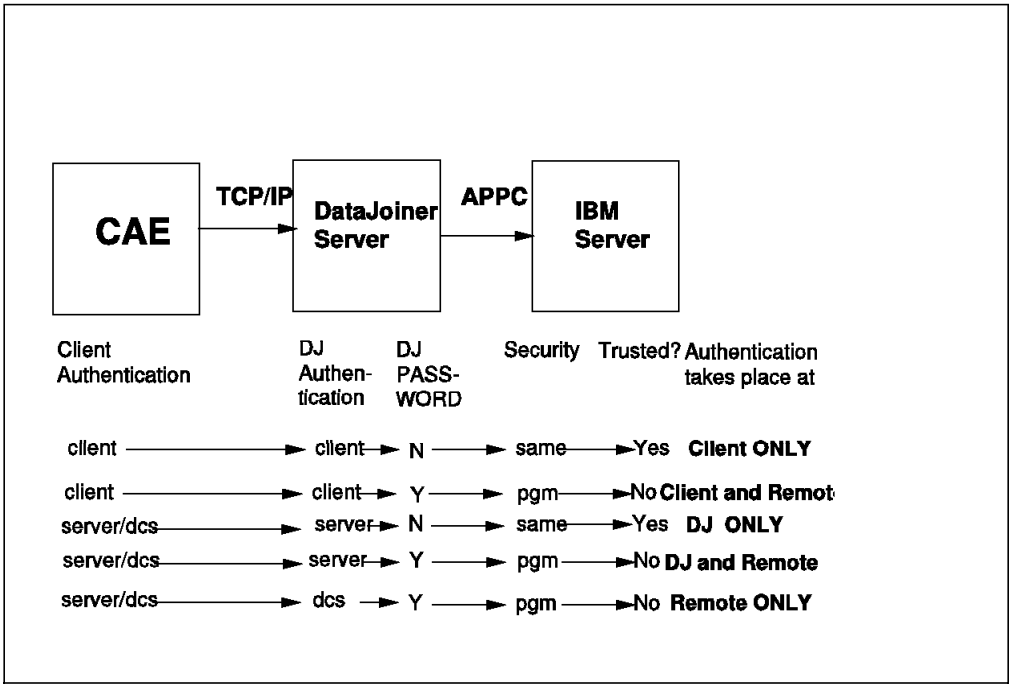


Figure 101. Security for Remote TCP/IP Clients to an IBM Server

6.3.4 Security for Local Clients Connecting to a Non-IBM Server

The following options are available for controlling local user security:

- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on *both* the DataJoiner workstation and the remote data sources.
- Users are authenticated on the remote data source *only*.

6.3.4.1 Authentication at DataJoiner Workstation Only

Because the client is running on the same machine as the server, there is no difference whether authentication is `client` or `server`. That is, authentication is to take place on the DataJoiner machine. (Authentication = `DCS` means that authentication is to take place at the remote data source.)

When the authentication at DataJoiner is set to either `CLIENT` or `SERVER`, the `authid` and `password` are validated by the AIX security system at the DataJoiner workstation when you connect to DataJoiner. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid` for that user/server pair and forwards that `authid` (if found) or the local `authid` to the remote data source.

When set up in this manner, the value for PASSWORD in SYSSERVERS must be set to `N` indicating that no password is to be sent to the remote data source because the remote data source does not validate passwords. The connection between DataJoiner and the remote data source is by means of TCP/IP; therefore, no security variable is to be set as there would be with APPC. The remote data source must be running in trusted-client mode.

6.3.4.2 Authentication at DataJoiner and Remote Data Sources

In order to achieve authentication at both DataJoiner and the remote data source, Authentication must be set to either CLIENT or SERVER. The user ID and password will be authenticated at DataJoiner when you attempt to connect. The value of PASSWORD in SYSSERVERS must be set to Y causing a password to flow to the remote data source. DataJoiner accesses SYSREMOTEUSERS to find the remote authid/password for that user/server pair, and if found, forwards the correct authid/password to the remote data source. If not found, the userid/password combination used to access DataJoiner is sent. The connection between DataJoiner and the remote data source is by means of TCP/IP, so there is no security parameter to be set as there would be with APPC. Because this scenario includes validation at the source, the source will not be running in trusted-client mode.

6.3.4.3 Authentication at Remote Data Source Only

In order to achieve authentication at the remote data source only, Authentication must be set to DCS. The user ID and password are not authenticated at DataJoiner when you connect. The value of PASSWORD in SYSSERVERS must be set to Y, causing a password to flow to the remote data source. DataJoiner accesses SYSREMOTEUSERS to find the remote authid/password for that user/server pair, and if found, forwards the correct authid/password to the remote data source. If they are not found, the userid/password combination used to access DataJoiner is sent. The connection between DataJoiner and the remote data source is by means of TCP/IP, so there is no security parameter to be set as there would be with APPC. Because this scenario includes validation at the remote data source, the source will not be running in trusted-client mode.

6.3.4.4 Summary, Local DataJoiner Clients Connecting to Non-IBM Server

Figure 102 summarizes the combinations for specifying security for local DataJoiner clients through to a non-IBM remote data source.

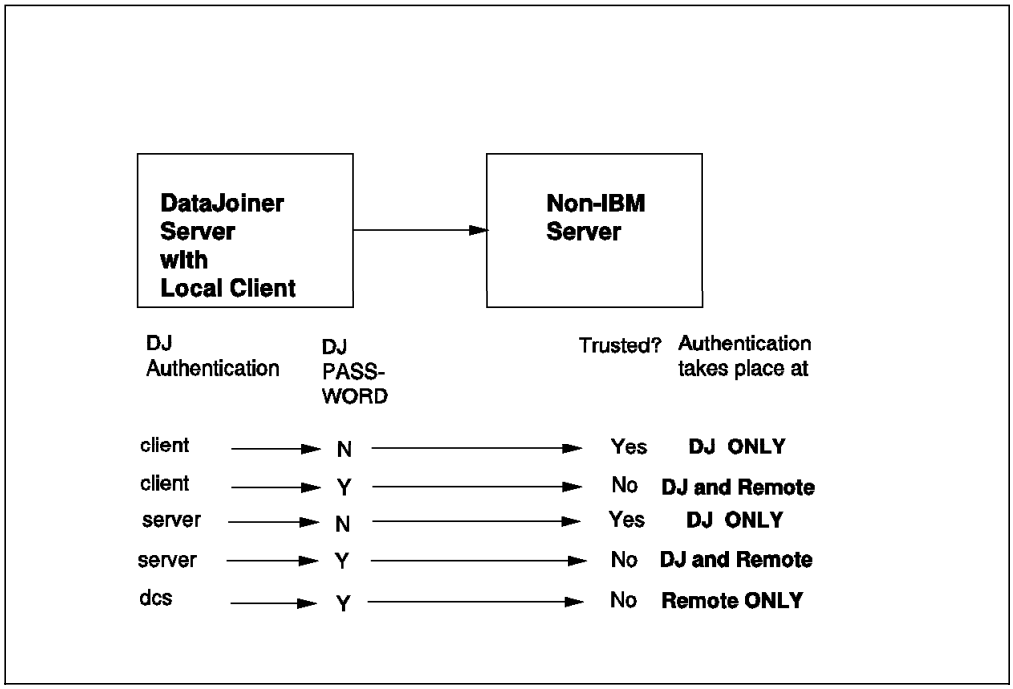


Figure 102. Security for Local DataJoiner Clients to a Non-IBM Server

6.3.5 Security for Remote APPC Clients Connecting to a Non-IBM Server

The following options are available for controlling remote user security:

- Users are authenticated on the client workstation *only*.
- Users are authenticated on *both* the client workstation *and* the remote data sources.
- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on *both* the DataJoiner workstation *and* the remote data sources.
- Users are authenticated on the remote data source *only*.

Because there is an extra APPC layer between the client and DataJoiner, you can specify a security classification at the APPC level as well. However, this layer of security is redundant and difficult to maintain. Therefore SECURITY = NONE is recommended at the client; it is not permitted on the connection to the remote data source.

6.3.5.1 Authentication at Client Workstation Only

When the authentication at the client is set to CLIENT, then authentication at DataJoiner must also be set to CLIENT. The authid and password are validated at the client when you login. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote authid for that user/server pair and forwards that authid (if found) or the local authid to the remote data source.

When set up in this manner, the value for PASSWORD in SYSSERVERS must be set to N indicating that no password is to be sent to the remote data source because the source does not validate passwords. The remote data source must be running in trusted-client mode.

6.3.5.2 Authentication at Client and Remote Data Sources

The client will be configured exactly as for authentication at DataJoiner only. (See 6.3.5.1, “Authentication at Client Workstation Only.”) However, slight differences apply to the DataJoiner workstation. Because we also wish to authenticate at the remote data source, PASSWORD in SYSSERVERS must be set to Y, indicating that a password is to be sent to the remote data source. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote authid and password for that user/server pair and forwards that authid/password (if found) or the local authid/password to the remote data source.

In this case APPC security between DataJoiner and the remote data source must be set to PGM, indicating that both user ID and password are to be sent. The remote data source is not running in trusted-client mode.

6.3.5.3 Authentication at DataJoiner Workstation Only

From the point of view of the client, authentication will be performed down the line, either at DataJoiner or the remote data source. Therefore authentication at the client can be set to either SERVER or DCS. Authentication at DataJoiner must be set to SERVER but PASSWORD must be set to N in SYSSERVERS. The remote data source must be running in trusted-client mode.

When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid` for that user/server pair and forwards that `authid` (if found) or the local `authid` to the remote data source.

6.3.5.4 Authentication at DataJoiner and Remote Data Sources

The client is configured exactly as for authentication at DataJoiner only. (See 6.3.1.1, “Authentication at DataJoiner Workstation Only” on page 150.) However, slight differences apply to the DataJoiner workstation. Because we also wish to authenticate at the remote data source, PASSWORD in SYSSERVERS must be set to `Y`, indicating that a password will be sent to the remote data source. When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid` and `password` for that user/server pair and forwards that `authid/password` (if found) or the local `authid/password` to the remote data source. The remote data source does not run in trusted-client mode.

6.3.5.5 Authentication at Remote Data Sources Only

From the point of view of the client, authentication is performed down the line, either at DataJoiner or at the remote data source. Therefore, authentication at the client can be set to either `SERVER` or `DCS`. Authentication at DataJoiner must be set to `DCS`, and PASSWORD must be set to `Y` in SYSSERVERS. Because there is no APPC connection between DataJoiner and the remote data source, there is no need to be concerned with APPC security. The remote data source must not be running in trusted-client mode.

When DataJoiner attempts a connection to a remote data source on your behalf, it accesses the SYSREMOTEUSERS table to find the remote `authid/password` for that user/server pair and forwards that `authid/password` (if found) or the local `authid/password` to the remote data source.

6.3.5.6 Summary, Remote APPC Clients Connecting to a Non-IBM Server

Figure 103 on page 159 summarizes the combinations for specifying security from remote APPC clients through to a non-IBM remote data source.

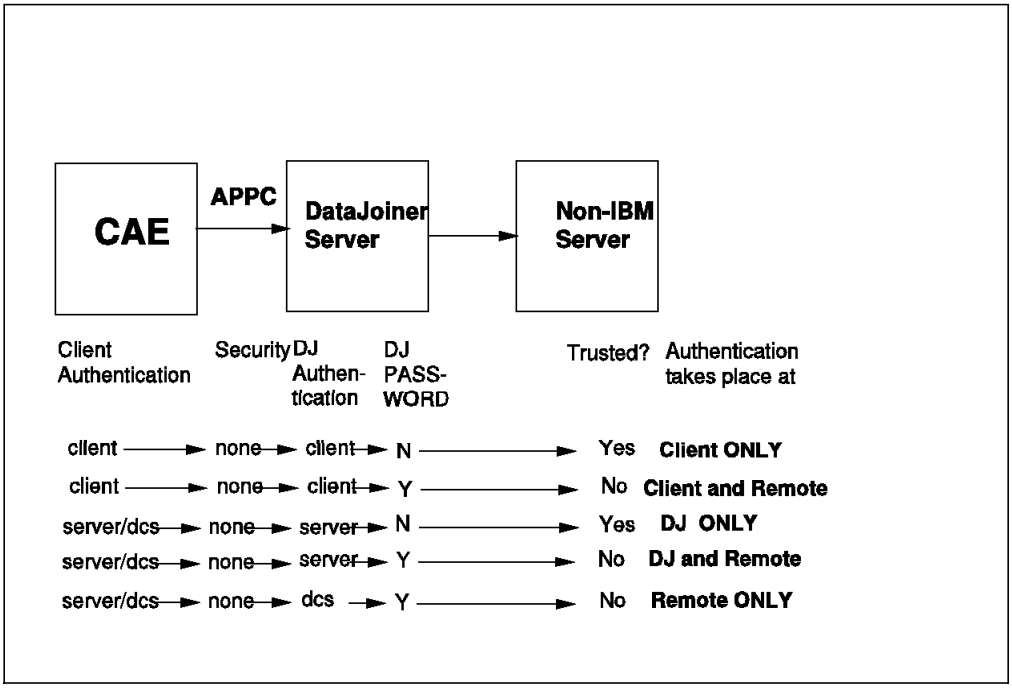


Figure 103. Security for Remote APPC Clients to a Non-IBM Server

6.3.6 Security for Remote TCP/IP Clients Connecting to a Non-IBM Server

The following options are available for controlling access by remote TCP/IP clients:

- Users are authenticated on the client workstation *only*.
- Users are authenticated on *both* the client workstation *and* the remote data sources.
- Users are authenticated on the DataJoiner workstation *only*.
- Users are authenticated on *both* the DataJoiner workstation *and* the remote data sources.
- Users are authenticated on the remote data source *only*.

The implementation is exactly as in 6.3.5, “Security for Remote APPC Clients Connecting to a Non-IBM Server” on page 157, except that no security is specified between the client and DataJoiner. TCP/IP has no security option as APPC does.

6.3.6.1 Summary, Remote TCP/IP Clients Connecting to a Non-IBM Server

Figure 104 on page 160 summarizes the combinations for specifying security from remote TCP/IP clients through to a non-IBM remote data source.

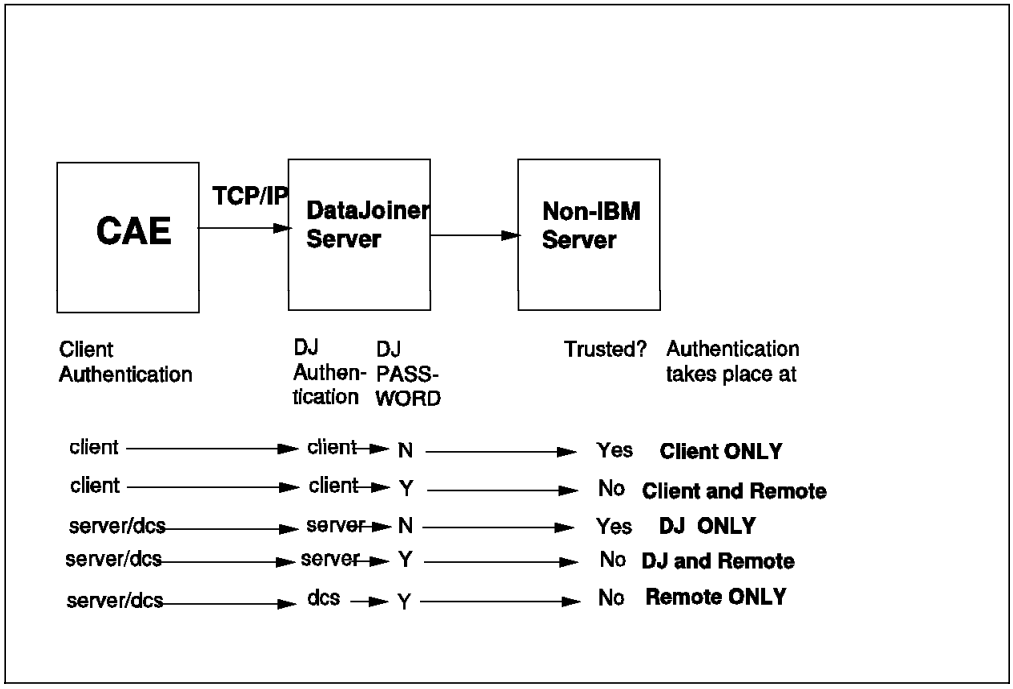


Figure 104. Security for Remote TCP/IP Clients to a Non-IBM Server

Chapter 7. Using DataJoiner

This chapter introduces the capabilities of DataJoiner. The purpose is to describe capabilities and considerations of using DataJoiner. The following topics are covered:

- Enabling users for access to local and remote data
- SQL considerations
- Using the passthrough facility.

7.1 Enabling Users for Access to Local and Remote Data

Regardless of the fact that data may be distributed among heterogeneous systems, users must have the perception that all data resides in the same data server. Enabling users for access to data through DataJoiner implies allowing them to code and execute SQL statements as if all data were on one system—DataJoiner.

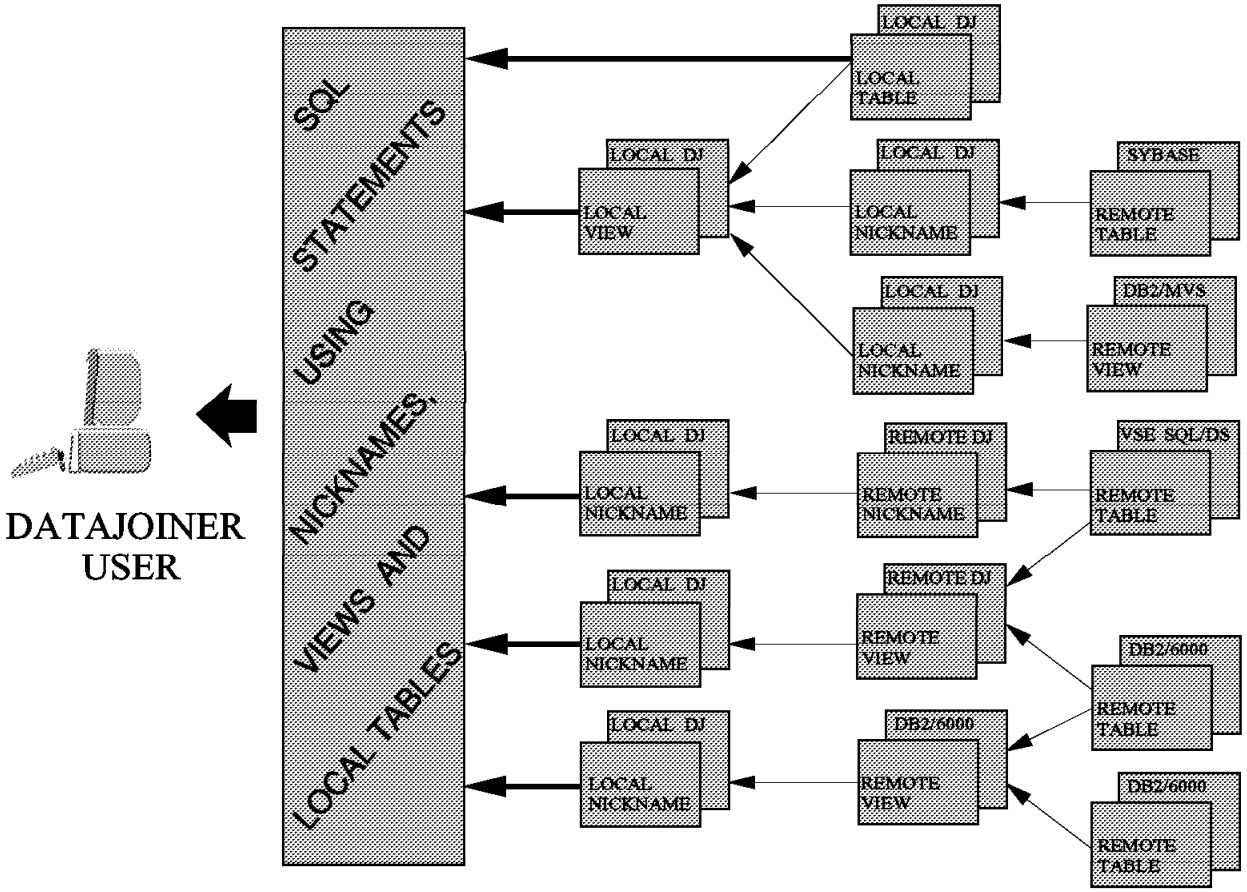


Figure 105. Providing Access to Data

To enable users to access data accessible by DataJoiner, follow these steps:

1. Set up the user or client as a DataJoiner user.
2. Grant connect privilege on a DataJoiner database to the user or client.

© Copyright IBM Corp. 1995

3. Create specific entries in SYSIBM.SYSREMOTEUSERS for each data source to which the user needs access.
4. Create nicknames for each remote table or view the user needs to access.
5. If required, create local views using the nicknames defined.
6. Create local tables to store information from remote data sources.
7. Grant privileges to a user or client on the nicknames, views, and local tables.
8. Ensure that the user or client has appropriate privileges on the remote tables or views at each remote data source.
9. Code and run SQL statements using local tables, nicknames, and views.

Table 5 provides information on the environment that is used in the examples throughout this chapter.

Table 5. Test Environment Definitions

Database Type	Database Name	Locally Known as	Creator and Table Name
DB2/VM	S34VMDB0	DB2VM	VMDBA.EMPLOYEES
DB2/VSE	S34VSDB1	DB2VSE	VSEDBA.IBM_EMPLOYEES
AIX DB2/6000	BOE6000	DB26KLOC	AIXDBA.EMPLOYEES
AIX DB2/6000	BOE6000	DB26KLOC	BOEUS1.COUNTRIES
DATAJOINER	DATAJOIN	DATAJOIN	
DATAJOINER (remote)	DBINST1	DBSJCDJ	

7.2 Multiple DataJoiner Databases and Instances

Many different scenarios are viable with DataJoiner. It is possible to create multiple DataJoiner databases in the same AIX system. Whether they are under the same DataJoiner *instance* or not, a user can be connected to only one database at a time. This does not imply a limitation, as any DataJoiner database may be defined as a *server* to the others. For practical purposes, consider the *local database* to be the one that a user is currently connected to. Any other data server (including another DataJoiner database) is then considered a *server* and must be locally defined as such.

The installation may also install DataJoiner in multiple AIX systems in the LAN. Some reasons for such a decision follow.

7.2.1 Disk Storage

When it is desirable to replicate data in local DataJoiner databases, one machine alone may not have enough disk storage to hold all the information to be replicated.

Sometimes DataJoiner may need to retrieve and temporarily hold large amounts of data to satisfy a given SQL request and disk storage must be sufficient to accommodate this data.

7.2.2 Security

Many security requirements must be met before a user or client has access to data:

- The user needs a valid user ID and password for an AIX system where DataJoiner is installed.
- The user must be explicitly authorized to connect to a DataJoiner database (provided no GRANT CONNECT ON DATABASE TO PUBLIC was issued).
- The user must be explicitly authorized to use a nickname, table, or view defined in DataJoiner (provided that no GRANT SELECT|INSERT|DELETE|UPDATE ON NICKNAME|*table-name*|*view-name* TO PUBLIC was issued).
- The user must have a valid user ID and password for the remote server or an entry in SYSIBM.SYSREMOTEUSERS must be made for that user.
- The user must be explicitly authorized to connect to the remote server.
- The user must be explicitly authorized to use a nickname, table, or view defined in the remote server (provided that no GRANT SELECT|INSERT|DELETE|UPDATE ON NICKNAME|*table-name*|*view-name* TO PUBLIC was issued).

One step beyond all those previously mentioned would be to *segregate* groups of users to DataJoiner databases that can be accessed only from specific remote servers.

7.2.3 Capacity

Processor speed and real memory availability limit the number of users and clients that can be connected to DataJoiner databases concurrently. To work around the limit, it may be advisable to distribute clients and users among DataJoiner databases running on different AIX systems in the LAN.

7.2.4 Connectivity

One DataJoiner database can access multiple data servers concurrently using different protocols and network paths, such as a connection to an MVS/ESA system through a 3174 control unit and a connection to a VSE/ESA system through a 3745 control unit. It is even possible to reach the same remote data server using both control units simultaneously. This capability makes it possible to determine which data shall be transmitted through which control unit.

Given two distinct AIX systems connected to hosts through different control units, it may be advantageous to create one DataJoiner database in each AIX system.

7.2.5 Cost

While DataJoiner does not require any additional product to access any member of the DB2 family (provided DRDA connectivity is already in place for DB2/MVS, SQL/DS, or DB2/400), specific code from different vendors is required to access other data sources. Enabling one DataJoiner database to access each of these data sources may be enough, as other DataJoiner databases can easily be configured to reach such remote data through the first one.

7.3 Steps to Enable Users for Use of DataJoiner

The following sections describe the procedures for enabling users to access data managed by DataJoiner.

7.3.1 Set Up the User or Client as a DataJoiner User

Enabling users to access DataJoiner is similar to enabling users to access DB2/6000. For remote clients, the corresponding CAE or SDK must be installed and customized. For local users, environment variables must be set in the user's profile. This step is thoroughly discussed in Chapter 3, "Configuration of DataJoiner for Clients" on page 29.

7.3.2 Grant Connect Privilege on a DataJoiner Database

Each user must be granted connect authority to access a DataJoiner database. To accomplish this step, a user with DBA authority on the database must connect to it and issue the commands as shown in Figure 106.

```
CONNECT TO datajoin USER djadmin USING djadmpw
GRANT CONNECT ON DATABASE TO djuser1
```

Figure 106. Grant Connect Privilege

7.3.3 Create Specific Entries in SYSIBM.SYSREMOTEUSERS

Whenever an SQL statement is run using a nickname, access to a remote data source is needed. If the user ID and password acknowledged by DataJoiner cannot be used at the remote data server, an entry must be made in SYSIBM.SYSREMOTEUSERS to specify which user ID and password are to be sent to each data server along with the SQL request. It may be easier for administrators to group DataJoiner users by assigning them one unique user ID at the remote data source. This lessens maintenance work, as SYSIBM.SYSREMOTEUSERS must be updated whenever passwords are changed in target systems. Figure 107 shows the general form of an SQL statement that may be used to insert a new row into SYSIBM.SYSREMOTEUSERS.

```
INSERT INTO SYSIBM.SYSREMOTEUSERS
(authid, server, remote_authid, remote_pw)
VALUES ('djuser1','DB2VM','VMUSER1','VMUSERPW')
```

Figure 107. Example of Updating SYSREMOTEUSERS

In this process,

authid is the SQL authorization ID the users will be using to connect to DataJoiner.

server is the name of the server where the remote authorization ID is defined. It must match an entry in the SERVER column of SYSSERVERS.

remote_authid is the SQL authorization ID at the data source to be used whenever the local user needs to access data from this specific data server.

remote-pw is the password of the remote authorization ID (*remote_authid*) at the data source.

7.3.4 Create Nicknames for Remote Tables and Views

Nicknames must be created in order to provide access to tables residing in any of the data servers accessed by DataJoiner. Nicknames map three-part names (location.owner.tablename) into two-part names (owner.tablename), thus making data locations transparent to the client, user, or application program. The database administrator may prefer to create all nicknames and then grant privileges on them to clients and users. The administrator should also consider creating nicknames under the same user ID to facilitate the work of application programmers and clients. To create a nickname, the following conditions must be met:

- The user creating the nickname must have CONNECT and CREATETAB privileges on a DataJoiner database.
- The user creating the nickname may need an entry in SYSIBM.SYSREMOTEUSERS referring to the specific data server where the table resides.
- The user creating the nickname must have SELECT privilege on the system catalog tables at the remote data source.
- The data server where the table resides must be running and must be connectable.

Figure 108 shows the general form of the CREATE NICKNAME statement.

```
CREATE NICKNAME vm_employees for db2vm.vmdba.employees
```

Figure 108. Create Nickname

7.3.5 Grant Privileges on the Nickname and Remote Data Objects

After a nickname is created, the end users perceive it exactly as they perceive a table or a view. However, when a nickname is used, at least two levels of authorization are checked:

- If the user ID using the nickname is not the one used when creating it, the system checks to make sure that privileges on the nickname are granted to that user or client.
- The system also checks to make sure that the user ID used to reach the data source (*remote_authid*) retains proper privileges on the base table or view for which the nickname was created. All authorizations must be in place in order to execute the SQL statement successfully. For instance, if a user wants to perform an SQL INSERT using a nickname, the user must have insert privilege on the nickname and the user ID used to access the remote data server must have an equivalent privilege on the base table.

In a network of interconnected DataJoiner databases, local nicknames may be created for nicknames defined in remote DataJoiner databases. This facility enables each DataJoiner database to act as a gateway to data sources for the other ones. In such a case, the user must have been granted privileges on all three sites: the local DataJoiner database, the remote DataJoiner database, and

the remote data server. Figure 109 on page 166 shows the general form of a typical SQL statement used to grant these privileges.

```
GRANT SELECT,INSERT ON vm_employees TO djuser1
```

Figure 109. Grant Access on Base Table

7.3.6 Create Local Views Using the Defined Nicknames

Views can be created using nicknames, making it easier for users and clients to access data, specially if unions, joins, or subqueries are necessary. The user who is running a SELECT statement using a view needs to have appropriate authorization on the view and on the base objects in the remote data servers. Figure 110 is an example of such a view.

```
CREATE VIEW totalemp (country_code, total_employees)
AS SELECT country_code, count(*)
FROM aixdba.vm_employees
GROUP BY country_code
```

Figure 110. Creating a View Using a Nickname

An important consideration pertaining to the use of nicknames defined for remote views is that the optimizer does not have the statistics for the remote tables and must use default statistics. Because of that, you should not indiscriminately create nicknames on remote views. This consideration does not apply for local views which depend on nicknames, if such nicknames reference remote tables.

7.3.7 Create Local Tables

Creating local tables on DataJoiner can avoid requests to remote data servers. Data for remote tables can be retrieved from sources periodically and held locally for queries. There can be several reasons to do this:

- If the source data is often accessed, but not frequently updated, a local replica of the remote base table can sometimes provide easier access and better performance.
- If the remote data server is often inaccessible, local tables may be essential to avoid delay.
- If the information is the result of a complex query that needs data from multiple data sources, a local table facilitates data assembly.
- If the remote data server does not provide good response times, a local table can improve response.

You can use a series of SQL statements to populate a local table from remote data, as shown in Figure 110.

```
. /home/djinst1/sqllib/db2profile
db2 "CONNECT TO datajoin USER aixdba USING aixdbapw"
db2 "DELETE FROM aixdba.localtable"
db2 "INSERT INTO dj_employees SELECT col1,col2,col3 FROM vm_employees"
db2 "INSERT INTO dj_employees SELECT col1,col2,col3 FROM vse_employees"
```

Figure 111. Storing Data Locally

In order to automate storage of local data, you can create a script with the commands in Figure 111. The script can be run by the AIX cron daemon at regularly scheduled intervals.

Use similar scripts to replicate information among the different data sources accessed by DataJoiner. Data can be easily copied from anywhere to anywhere. The easiest way is to copy whole tables, but you may want to copy subsets of information, coding SQL statements appropriately. One major concern about using this technique to replicate data is that the AIX system will not run the script if there is a power failure or if the system is down. Moreover, if the connection to a given host is unavailable at the time when the script is scheduled to run, the script will fail and no action will be taken by the AIX system, unless the script is coded accordingly.

7.4 Test Case

This test case exemplifies what has to be done in order to provide DataJoiner users and clients with access to remote data. Figure 112 on page 168 illustrates the environment for the test case.

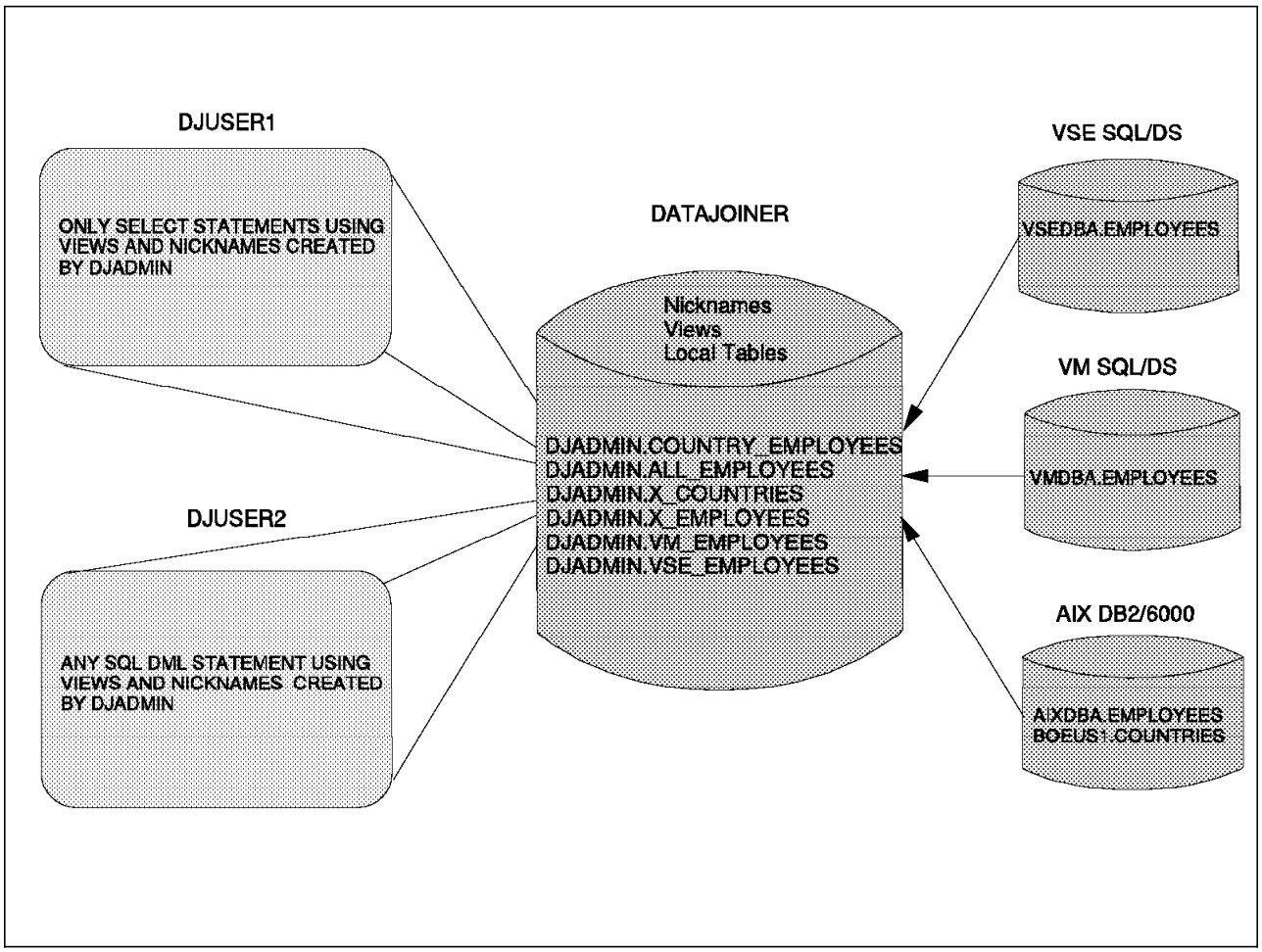


Figure 112. Test Case

The test case shown in Figure 112 is based on the following assumptions:

- User VMDBA creates a table named VMDBA.EMPLOYEES on a VM SQL/DS (DB2/VM) database called S34VMDB0. The server name DB2VM corresponds to this database in SYSIBM.SYSSERVERS.
- User VSEDBA creates a table named VSEDBA.EMPLOYEES on a VSE SQL/DS (DB2/VSE) database called S34VSDB1. The server name DB2VSE corresponds to this database in SYSIBM.SYSSERVERS.
- User BOEUS1 creates a table named AIXDBA.COUNTRIES on a DB2/6000 database called DB26000. The server name DB26KLOC was assigned to this database.
- User AIXDBA also creates a table named AIXDBA.EMPLOYEES on database DB26000.
- Users DJADMIN, AIXDBA, and BOEUS1 are the only users who have CREATETAB authority on the local DataJoiner database called DATAJOIN.
- User DJUSER1 has only CONNECT authority on database DATAJOIN and wishes to access information from the tables created by VMDBA, VSEDBA, AIXDBA, and BOEUS1. DJUSER1 needs to run queries that access the base tables directly, but also desires to access some information through views that consolidate information from multiple data sources. DJUSER1 cannot update any information on remote data sites.

- User DJUSER2 has CONNECT authority on all databases, as the DJUSER2 user ID and corresponding password are exactly the same in all systems. DJUSER2 wishes to run queries that will select, insert, delete, and update data on VMDBA.EMPLOYEES, VSEDBA.EMPLOYEES and AIXDBA.EMPLOYEES. DJUSER2 also needs the select privilege on BOEUS1.COUNTRIES.

In order to meet the needs of DJUSER1 and DJUSER2, the following actions must be taken:

1. DJADMIN must create nicknames for all tables created on remote database servers. To do so, DJADMIN needs prior authorization on the system catalog tables on the remote servers.

   Issue the following SQL statements:

   ```
   CREATE NICKNAME vm_employees FOR db2vm.vmdba.employees

   CREATE NICKNAME vse_employees FOR db2vse.vsedba.employees

   CREATE NICKNAME x_employees FOR db26kloc.aixdba.employees

   CREATE NICKNAME x_countries FOR db26kloc.boeus1.countries
   ```

2. Upon successful creation of the nicknames above, DJADMIN must issue SQL commands to create distributed views using the nicknames.

   Issue the following SQL statements:

   ```
   CREATE VIEW all_employees (name,telephone) AS
   SELECT name,country_code FROM vm_employees UNION ALL
   SELECT name,country_code FROM vse_employees UNION ALL
   SELECT name,country_code FROM x_employees

   CREATE VIEW country_employees (name, country_name, telephone) as
   SELECT t1.name, t2.country_name, t1.telephone
   FROM all_employees t1, x_countries t2
   WHERE t1.country_code = t2.country_code
   ```

3. User DJADMIN must provide proper authorizations to DJUSER1 and DJUSER2.

   Issue the following SQL statements:

   ```
   GRANT SELECT ON all_employees TO djuser1

   GRANT SELECT ON country_employee TO djuser1

   GRANT SELECT ON vse_employees TO djuser1,djuser2

   GRANT SELECT ON vm_employees TO djuser1, djuser2

   GRANT SELECT ON x_employees TO djuser1, djuser2

   GRANT SELECT ON x_countries TO djuser1, djuser2

   GRANT DELETE, UPDATE, INSERT ON vse_employees TO djuser2

   GRANT DELETE, UPDATE, INSERT ON vm_employees TO djuser2

   GRANT DELETE, UPDATE, INSERT ON x_employees TO djuser2
   ```

4. User AIXDBA must update SYSIBM.SYSREMOTEUSERS so that DJUSER1 can reach the remote data sources, because user ID DJUSER1 was not defined in all systems involved. Whenever DJUSER1 needs to access data from the remote servers, other user IDs and passwords will then be used. To update SYSIBM.SYSREMOTEUSERS, a series of SQL statements can be used.

   Issue the following SQL statements:

```
INSERT INTO SYSIBM.SYSREMOTEUSERS
(authid,server,remote_authid,remote_pw)
VALUES ('DJUSER1','DB2VM','DJREMVM','Y123R4H')

INSERT INTO SYSIBM.SYSREMOTEUSERS
(authid,server,remote_authid,remote_pw)
VALUES ('DJUSER1','DB26KLOC','DJLOCAIX','ZPTZ83W')

INSERT INTO SYSIBM.SYSREMOTEUSERS
(authid,server,remote_authid,remote_pw)
VALUES ('DJUSER1','DB2VSE','DJREMVSE','XPTO71W')
```

5. Even though DJUSER2, DJREMVM, DJLOCAIX, and DJREMVSE are user IDs that have connect authority on databases DB2VM, DB26KLOC, and DB2VSE, as appropriate, they need specific authorization to run queries against the base tables on each data server. This way, users VMDBA, VSEDBA, and AIXDBA must each provide proper authorizations.

   VMDBA needs to grant the following privileges:

```
GRANT SELECT ON employees TO djremvm, djuser2
GRANT INSERT,DELETE,UPDATE ON employees to djuser2
```

   VSEDBA needs to grant the following privileges:

```
GRANT SELECT ON employees TO djremvse, djuser2
GRANT INSERT,DELETE,UPDATE ON employees TO djuser2
```

   AIXDBA needs to grant the following privileges:

```
GRANT SELECT ON employees to djlocaix, djuser2
GRANT INSERT,DELETE,UPDATE ON employees TO djuser2
```

   BOEUS1 needs to grant the following privileges:

```
GRANT SELECT ON countries to djlocaix
GRANT SELECT ON employees to DJUSER2
```

6. User DJUSER1 and DJUSER2 can now code and run SQL commands using the views and nicknames.

7.5 Capabilities

Using the test case described in 7.4, “Test Case” on page 167 and shown in Figure 112 on page 168 as a basis for the examples, consider the following SQL statements, which illustrate the new capabilities provided by DataJoiner.

7.5.1 SQL Distributed Queries by Means of Subqueries

The query below is intended to retrieve information from the database server DB26KLOC in order to get the corresponding country_code for China. It then accesses the database server DB2VM, substituting the subquery for the value previously retrieved:

```
SELECT name, telephone
FROM DJADMIN.vm_employees
WHERE country_code IN
(SELECT country_code
FROM DJADMIN.x_countries
WHERE country_name = 'CHINA')
```

7.5.2 SQL Distributed Queries by Means of Set Operators

DataJoiner supports three set operators:

- UNION

  This set operator combines the rows that satisfy any of two or more SELECT statements.

- EXCEPT

  This set operator is used to retrieve those rows that satisfy the first SELECT statement but not the second.

- INTERSECT

  This set operator is used to retrieve those rows that satisfy both SELECT statements.

  All three set operators may have the ALL operand to indicate that duplicate rows are not to be removed from the result, thus eliminating the need for an extra sort.

The query below retrieves all employee names and country codes that are present in both the VM_EMPLOYEES and VSE_EMPLOYEES tables, even though the first table resides in server DB2VM and the second one in DB2VSE.

```
SELECT name, country_code
FROM vm_employees
INTERSECT
SELECT name, country_code
FROM vse_employees
```

7.5.3 SQL Distributed Query by Means of Relational Joins

A relational join produces a result table whose rows contain a combination of columns retrieved from two or more tables. Conditions should always be specified to limit the size of the resulting set of rows.

The query below combines employee names and their corresponding country names by comparing country codes present in both tables. Here, table VM_EMPLOYEES resides in server DB2VM and table X_EMPLOYEES resides in server DB26KLOC.

```
SELECT t1.name, t2.country_name
FROM DJADMIN.vm_employees t1, DJADMIN.x_countries t2
WHERE t1.country_code = t2.country_code
```

7.5.4 SQL INSERT Statements Using Distributed Functions

Statements other than SELECT statements may benefit from the distributed capabilities introduced by DataJoiner. The query below shows an INSERT statement that retrieves information from one database and inserts it into another:

```
INSERT INTO DJADMIN.vse_employees
(name,country_code,telephone,empnum)
SELECT name,country_code,telephone,empnum
FROM DJADMIN.vm_employees
WHERE country_code=631
```

7.5.5 SQL DELETE Statements Using Distributed Functions

The DELETE statement below requires that a table residing in DB26KLOC be read and the appropriate rows from a table residing in server DB2VSE then be deleted:

```
DELETE FROM DJADMIN.vse_employees
WHERE country_code in
(SELECT country_code
FROM DJADMIN.x_countries
WHERE country_code between 200 and 500)
```

7.5.6 SQL UPDATE Statements Using Distributed Functions

The UPDATE statement below requires that a table residing in server DB2VSE be read in order to calculate the average function. The salary of any employee that is below the calculated average is then increased by sixty percent. Note that the subquery is run against server DB2VSE while the UPDATE operation is done on server DB2VM. The full statement is as follows:

```
UPDATE DJADMIN.vm_employees
SET salary = salary * 1.6
WHERE mgr='N'
AND salary <
(SELECT AVG(salary)
FROM DJADMIN.vse_employees)
```

7.6 SQL Considerations

This section provides information on how DataJoiner handles various SQL statements on different data sources. A capability of SQL may be supported on one platform but not on another. If the capability is supported on both, it may still yield different results or may be supported in a different way. There are also some differences in data types and how they are handled. Differences exist not only between IBM and non-IBM data sources, but even among the DB2-family database management systems. DataJoiner must handle these differences consistently.

7.6.1 Compatibility

Whenever you are running queries using DataJoiner, remember that the various data servers involved may not support the same SQL dialect. DataJoiner compensates for lack of SQL support in certain data sources even if the function is not supported by the server. On the other hand, DataJoiner supports DB2/6000 SQL DML only. For example, if an SQL query is run using a SYBASE-specific or ORACLE-specific SQL statement, the DataJoiner SQL parser will not understand it and the request will fail. Whenever an SQL statement contains column or scalar functions that are not supported by the remote data server, DataJoiner must retrieve the whole table and perform the function locally. Such retrieval may lead to performance problems because of the great amounts of data transmitted. Because DataJoiner uses a ″push-down″ strategy to run queries, that is, DataJoiner tries to run query predicates as close as

possible to the data source, you should always try to code SQL statements that are supported by all data servers involved. For example, even though all vendors support column functions, each vendor implements different sets of scalar functions.

7.6.2 The DB2 Family

When porting applications across environments or when porting an existing application to use DataJoiner, have the SQLFRED document—*SQL Formal Register of Extensions and Differences*—available. That document is intended to help application developers in designing, implementing, and migrating applications using the DB2 family of products. As mentioned, DataJoiner uses the same SQL dialect as DB2/6000 Version 1.

7.7 Using the Passthrough Facility

The passthrough facility allows DataJoiner users and clients to send SQL statements directly to the data source. This way, users may then use any SQL statement that is supported by the data server even though it may not be supported by DataJoiner. Figure 113 illustrates the passthrough facility and its relationship to the user or client, to DataJoiner, and to the data source.

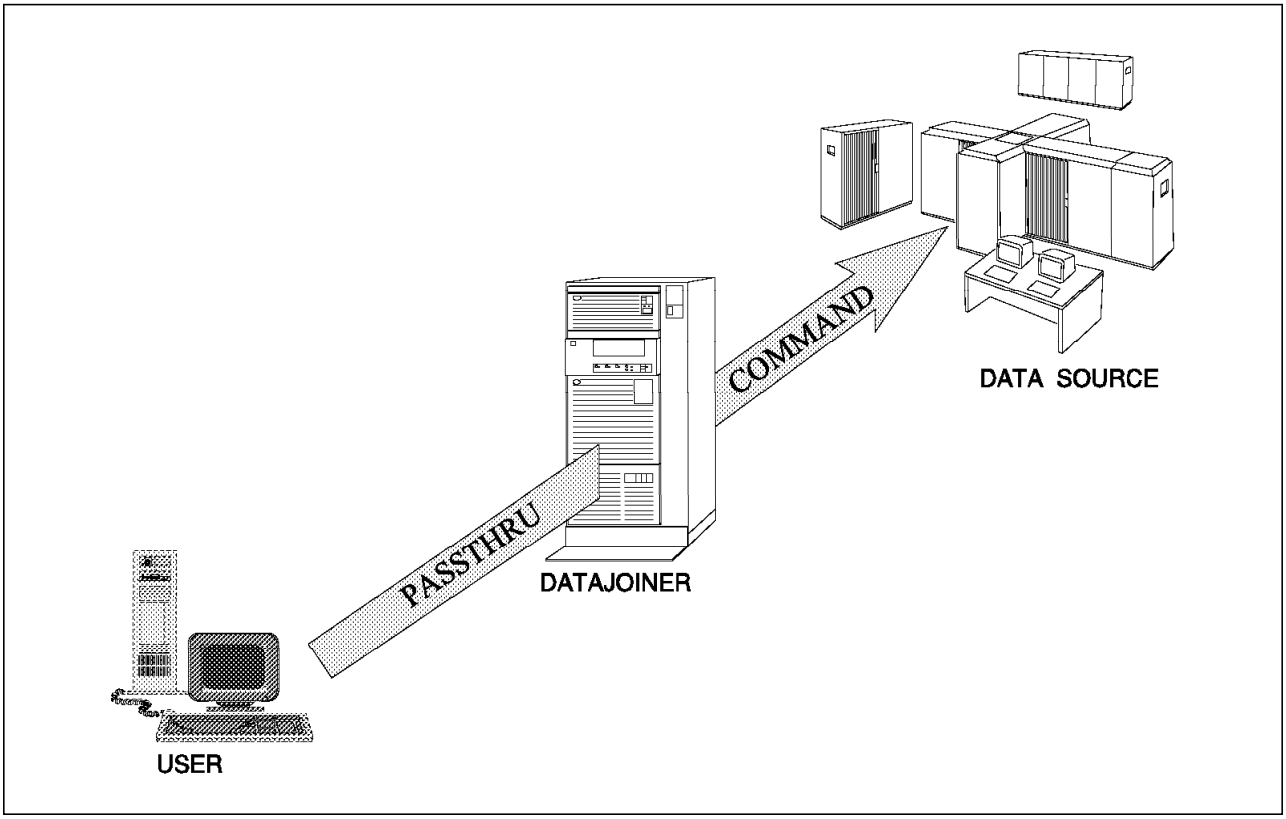


Figure 113. Using the Passthrough Facility

As an example, the passthrough facility is needed for the `create synonym` command, because only SQL/DS (DB2 for VM and VSE) and DB2/MVS support that statement. Figure 114 on page 174 is an example of using the passthrough facility to run SQL statements from DataJoiner on systems with DB2 for VM and VSE.

```
SET PASSTHRU DB2VM
CREATE SYNONYM VMSYN  FOR SYSTEM.SYSCATALOG
SET PASSTHRU DB2VSE
CREATE SYNONYM VSESYN FOR SYSTEM.SYSCOLUMNS
```

Figure 114. Example of Using SET PASSTHRU Statements

The SET PASSTHRU RESET statement will end the passthrough session.

The SET PASSTHRU statement can only be issued dynamically. If it is desirable to imbed this command in an application program, use either the PREPARE, EXECUTE, or EXECUTE IMMEDIATE SQL statements.

Even though the SET PASSTHRU statement will force a connection directly to the data source, there may be restrictions in regard to the protocol being used to access the remote server. For example, when using SET PASSTHRU to connect to DRDA application servers, be aware that DRDA itself does not support the full set of SQL statements, thus imposing a few extra restrictions.

Chapter 8. DataJoiner Capacity and Performance

This chapter discusses some ways to tune your DataJoiner system including LU 6.2 sessions, DB2 parameters, and AIX processes. The chapter also covers some ways to help you find out what your DataJoiner system is doing. These include DB2 trace facilities (extended for DataJoiner), DB2 monitoring facilities (extended for DataJoiner), and the DataJoiner Explain tool.

8.1 System Tuning

This section looks at ways to tune your DataJoiner system in the following areas:

- LU 6.2 sessions and DB2 parameters
- AIX processes.

8.1.1 LU 6.2 Sessions and Related DB2 Parameters

Every DataJoiner user who wishes to access a DRDA-attached data source requires its own session between DataJoiner and the DRDA data source. If there are more concurrent users than available sessions between DataJoiner and the DRDA data source, some users have to wait for sessions to become available, causing them to experience poor response time.

DataJoiner and the DRDA data source both specify the maximum number of available sessions. When a connection is established between DataJoiner and a DRDA data source, the two partners negotiate the number of sessions to be available between them. This figure will be the lower of the session limit values specified at DataJoiner and the DRDA data source.

To see the number of sessions defined for DataJoiner, use SMIT. Login as root and enter

```
smit sna
```

at the command line. Select the following options from the SNA Server/6000 V2 window:

Configure SNA Profiles

Advanced Configuration

Sessions

LU 6.2

LU 6.2 Mode

Change/Show a Profile.

At the LU 6.2 Mode Profile Name window, enter the name of your LU 6.2 Mode profile. You can use **PF4=List** to see a list of the available options. This profile is described in section “LU 6.2 Mode” on page 88. When you have made your selection, press Enter. Figure 115 on page 176 shows the Change/Show LU 6.2 Mode Profile window.

© Copyright IBM Corp. 1995

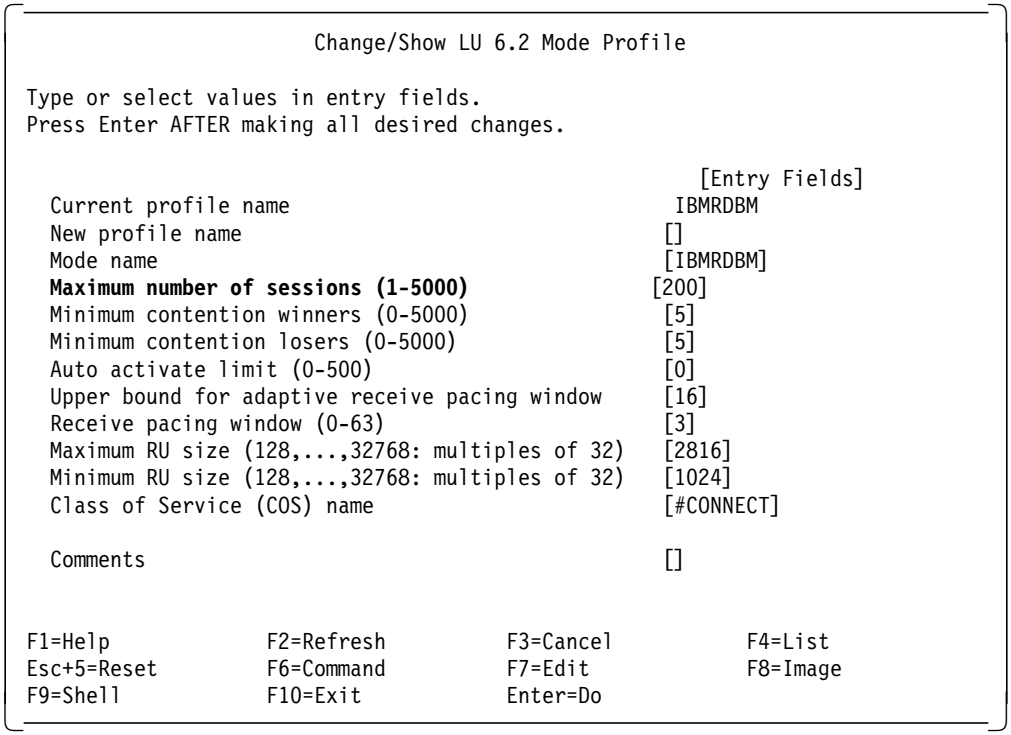


Figure 115. Change/Show LU 6.2 Mode Profile

The value for Maximum number of sessions (1-5000) is the maximum number of sessions defined at DataJoiner. In our system, this value is 200.

To find the number of sessions defined for the DRDA data source, look in the following places:

DB2/MVS	VTAM APPL definition for DB2 and DB2′s SYSIBM.SYSLUMODES table
SQL/DS	AVS LU definition for DB2
DB2/VSE	CICS LU definition for DB2
DB2/400	CRTMODD definition for DB2

As an example, we consider the DB2/MVS definitions for sessions. Figure 116 on page 177 shows the VTAM APPL statement for our DB2/MVS system.

```
SCLUDB3A  APPL ACBNAME=SCLUDB3A,  APPLID FOR DB2 V3
               APPC=YES,
               ATNLOSS=ALL,       NEW PARM WITH DB2 V310
               AUTH=(ACQ),        ACCESS TO VTAM FUNCTIONS
               AUTOSES=10,
               DMINWNL=25,
               DMINWNR=25,
               DSESLIM=50,       MAX NUMBER OF SESSIONS IN USE
               EAS=509,
               ENCR=NONE,
               MODETAB=AGWTAB,
               PARSESS=YES,
               SECACPT=ALREADYV,
               SONSCIP=NO,
               SRBEXIT=YES,      SRB PROCESSING IN EXIT ROUTINES
               SYNCLVL=SYNCPT,   NEW PARM FOR DB2 V310
               VERIFY=NONE,
               VPACING=2,
               VTAMFRR=NO
```

Figure 116. VTAM APPL Definition for DB2/MVS

DSESLIM=50 specifies that the session limit for DB2/MVS is 50. When session negotiation occurs between our DataJoiner and our DB2/MVS, the resulting number of available sessions is 50, which is the lower of the two session limits.

The DB2/MVS SYSIBM.SYSLUMODES table contains the following columns:

- LUNAME
- MODENAME
- CONVLIMIT
- AUTO.

Any entries in this table override the DSESLIM value in the VTAM APPL definition. Our SYSIBM.SYSLUMODES table is empty, so we have a session limit of 50 between DataJoiner and DB2/MVS.

8.1.2 AIX Process Monitoring

Every DataJoiner client that wishes to connect to a data source needs its own process under the DataJoiner instance owner. AIX sets the maximum number of processes for each user as a system default. If you have many clients who wish to use DataJoiner at the same time, you must ensure that the maximum number of processes allowed per user is high enough to accommodate them.

To determine the number of running processes, issue the following command:
`ps -ef | grep` **instance_owner** `| wc -l`

Substitute the name of your DataJoiner instance owner for **instance_owner**.

This will return a single number, which is the number of processes currently running under the instance_owner. This number may be slightly inaccurate as the instance_owner name may show up for more than one process.

The default number of processes allowed per user in AIX is 40. To determine the current maximum number of running processes allowed per user, use SMIT. Login as root and enter

```
smit
```

at the command line. Select the following options from the System Management window:

System Environments

Change / Show Characteristics of Operating System.

Figure 117 shows the Change / Show Characteristics of Operating System window.

```
                Change / Show Characteristics of Operating System

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

                                                      [Entry Fields]
  Maximum number of PROCESSES allowed per user       [300]
  Maximum number of pages in block I/O BUFFER CACHE  [40]
  Maximum Kbytes of real memory allowed for MBUFS    [2048]
  Automatically REBOOT system after a crash           false
  Continuously maintain DISK I/O history              true
  HIGH water mark for pending write I/Os per file    [0]
  LOW water mark for pending write I/Os per file     [0]
  Enable memory SCRUBBING                             false
  Amount of usable physical memory in Kbytes          131072
  Primary dump device                                 /dev/hd7
  Secondary dump device                               /dev/sysdumpnull
  Error log file size                                 1048576
  State of system keylock at boot time                normal
  Size of data cache in bytes                         32K
  Size of instruction cache in bytes                  32K

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 117. Change / Show Characteristics of Operating System

Ensure that the value specified for maximum number of processes allowed per user is large enough for your number of DataJoiner users. If you need to increase this value, type in the new value and press Enter; the change will take effect immediately. If you decrease this value, however, the change will take effect at the next system boot time.

8.2 System Monitoring

This section covers the following topics:

- DB2 trace facilities (extended for DataJoiner)
- The DataJoiner Explain tool
- DB2 monitoring facilities (extended for DataJoiner).

8.2.1 Trace Facilities

The Independent Trace Facility has been extended by DataJoiner to trace remote data sources. This can be very useful in problem diagnosis where the error has occurred at the remote data source and more information is required to diagnose the problem. The formatted trace output will contain the commands executed at the data source and the native return codes. Figure 118 shows a sample of the formatted trace output containing an insert into a Sybase table.

```
207     DB2 fnc_data      Middleware gateway   open-client execI (1.35.49
        pid 29452; cpid 20747; time 0; trace_point 2
        494e 5345 5254 2049 4e54 4f20 6977 646a INSERT INTO iwdj
        3031 2e74 7063 635f 6869 7374 6f72 7928 01.tpcc_history(
        685f 635f 6964 2c68 5f63 5f64 5f69 642c h_c_id,h_c_d_id,
        685f 635f 775f 6964 2c68 5f64 5f69 642c h_c_w_id,h_d_id,
        685f 775f 6964 2c68 5f64 6174 652c 685f h_w_id,h_date,h_
        616d 6f75 6e74 2c68 5f64 6174 6129 2056 amount,h_data) V
        414c 5545 5328 2832 3129 2c28 3929 2c28 ALUES((21),(9),(
        3229 2c28 3929 2c28 3229 2c27 3139 3935 2),(9),(2),'1995
        3034 3231 2031 343a 3232 3a33 373a 3030 0421 14:22:37:00
        3027 2c28 312e 3030 3030 3030 652b 3030 0',(1.000000e+00
        292c 2730 6d41 6567 6c4b 5261 6820 2020 ),'0mAeglKRah
        2062 6964 7867 7169 2020 2027 2900 4141  bidxgqi   ').AA
        312e 635f 6363 7265 6464 6974 5f5f 6c69 1.c_ccreddit__li
        6d6d 2c41 4162 616c 6179 7464 5f        mm,AAbalaytd_

208     DB2 fnc_retcode   Middleware gateway   open-client execI (1.33.49
        pid 29452; cpid 20747; time 0; trace_point 254
        return_code = 000000 = 0
```

Figure 118. DB2TRC Formatted Trace Output

Figure 119 on page 180 shows a sample of the formatted trace output containing an error message from an Oracle data source.

```
421     DB2 non-fatal_err Middleware gateway   SQLNET error (1.4.49.159)
        pid 19715; cpid 8952; time 0; trace_point 1
        4f52 412d 3031 3534 373a 2066 6169 6c65 ORA-01547: faile
        6420 746f 2061 6c6c 6f63 6174 6520 6578 d to allocate ex
        7465 6e74 206f 6620 7369 7a65 2035 2069 tent of size 5 i
        6e20 7461 626c 6573 7061 6365 2027 5553 n tablespace 'US
        4552 5327 0a00 f1e8 2019 0004 0000 0002 ERS'.... .......
        0000 0008 2ff7 f1c8 6e20 7365 0000 0001 ..../...n se....
        735f 6c61 2ff7 f1f8 2019 0004 201b 2794 s_la/... ... .'.
        0000 0000 0000 0000 0000 0000 0000 0000 ................
        0000 0000 2ff7 f1e8 dead beef dead beef ..../...........
        dead beef 2ff7 f208 0000 0001 dead beef ..../...........
        0000 0000 204d 470c 2019 0004 0000 0003 .... MG. .......
        0000 0002 2ff7 f228 0000 0000 0000 0000 ..../..(........
        201b 2794 2ff7 f288 2019 0004 d1d6 ecd8  .'./... .......
        2ff7 f2ac 2ff7 f268 4224 4828 d1d7 6760 /.../..hB$H(..g
        2009 c250 2ff7 f288 2019 57f8 d1b5 e038  ..P/... .W....8
        2009 c0b0 2ff7 f268 2019 57f8 0000 da58  .../..h .W....X
        2009 c250 2ff7 f2a8 2019 0004 d1b5 e038  ..P/... ......8
        0000 0000 0000 0000 dead beef 0000 da74 ...............t
        0000 0000 2ff7 f2c0 0000 0000 0000 0000 ..../...........
        0000 0024 0000 da34 c000 e4c4 0000 0003 ...$...4........
        2009 c250 2ff7 f2e8 0000 0000 d1b5 e038  ..P/..........8
        0000 001c 2ff7 f2b8 c000 e4e0 0000 df60 ..../..........
        2009 c0b0 0000 000c 0000 0000 d1b5 bf4c  ..............L
        0000 0000 2ff7 f2f8 2019 57f8 0000 0003 ..../... .W.....
        0000 0000 2ff7 f320 0000 0000 0000 0000 ..../.. ........
        0000 0020 0000 df40 c000 e9cc c000 e9b0 ... ...@........
        ffff fff4 0000 000c 0000 0000 201b 2762 ............ .'b
        0000 0001 0000 0000 2019 57f8 0000 0000 ........ .W.....
        2009 c0b0 2ff7 f348 2019 57f8 d1b5 bf4c  .../..H .W....L
        204d 470c 2ff7 f348 2019 0004 0104 3199  MG./..H .....1.
        ffff fff4 2009 c0c4 ffff fffc 2ff7 f320 .... ...../..
        ffff fffc 2ff7 f384 0000 0000 0000 0003 ..../...........
```

Figure 119. DB2TRC Formatted Trace Output Containing an Error Message

To start the Independent Trace Facility for DataJoiner, issue the following command:

```
db2trc on -m *.*.49-50.*
```

Now, run the command that you wish to trace. When the command has completed, issue the following command:

```
db2trc fmt > trc.out
```

Replace **trc.out** with the name of the file where you want the formatted trace to be written to. To turn tracing off, issue the following command:

```
db2trc off
```

The independent trace facility is described in more detail in *IBM DataJoiner Administration*.

8.2.2 Using the DataJoiner Explain Tool

The Explain tool is a productivity aid that interprets the access packages for the SQL statements. A complete description of the Explain tool is given in *IBM DataJoiner Administration*. The Explain tool returns a step-by-step description of the access strategy created by the DataJoiner global optimizer. The SQL statements that can be analyzed with the Explain tool are SELECT, UPDATE, INSERT or DELETE. However, the SELECT SQL statement is the prime candidate for the Explain analysis.

The Explain tool can be used to evaluate an existing performance problem. With the Explain tool you can analyze such things as:

- Detailed sort requirements
- Order of access of multiple table joins
- Correct usage of an index
- The locking strategies.

The Explain tool can also be used to prototype changes to an application.

The Explain tool reads a stored package from the catalog, interprets the contents of one or more sections, and writes the resulting analysis to an output file in the form of text.

To run Explain, do the following:

1. Run the statistics utilities at all applicable data sources.
2. Run the program or statement to be explained so that DataJoiner can gather current statistics from the data sources. This must be done before you run the Explain tool because, when Explain runs, it interrogates only the global catalogs to formulate an access strategy and does not access the data sources.
3. Run Explain as described in "Using Explain for Packages" or as described in "Using Explain for Dynamic SQL Statements" on page 183.

You can use Explain in two ways. You can run Explain for packages that are already bound to the database, or you can use Explain to analyze dynamic SQL statements. The following sections describe how to use these techniques.

Using Explain for Packages: To analyze the access strategy for a package, do the following:

1. Precompile your application program with the PREP command, using the BINDFILE and PACKAGE options.
2. Run the db2expln command; its syntax is shown in Figure 120 on page 182.

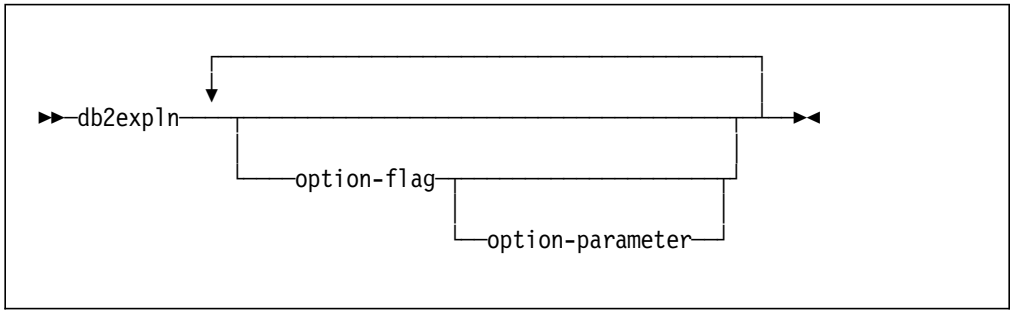


Figure 120. Syntax for the db2expln Command

Table 6 shows the option flags for the db2expln command.

Table 6. Option Flags for the db2expln Command

Option	Option Parameter	Description
-c	Creator	User ID of the package creator.
-d	Database name	Name of the database that contains the package or packages.
-h	Help	An option requesting help for the option parameters.
-o	Output file	The name of the file that will contain the output from the Explain tool.
-p	Package name	The name of the package containing the SQL statements to be explained.
-s	Section number	Section number within the package to be explained. Section numbers are found in the catalog table, SYSIBM.SYSSTMT. If 0 is specified, all sections in the packages are to be explained.

As an example, in our environment we implement a program that performs read-only transactions. We precompile this program, giving the name of ORD to its package. To run the Explain tool on this package, and find the access strategy for the first section of the package, we issue the following command:

```
db2expln -d tpcc -p ORD -c djinst1 -s 1 -o ord.expln
```

Running this command produces the following report:

```
DB2/6000 Version 1.1.0, 5622-044 (c) Copyright IBM Corp. 1991, 1993
Licensed Material - Program Property of IBM
IBM DATABASE 2 AIX SQL Explain Function

******************** PACKAGE ************************************
```

```
Package Name = DJINST1.ORD
      Prep Date = 1995/05/01
      Prep Time = 12:28:00:045

-------------------- SECTION ---------------------------------------
Section = 1


SQL Statement:
  SELECT C_ID, C_FIRST, C_MIDDLE, C_LAST, C_BALANCE
  FROM CUSTOMER
  WHERE C_LAST = :c_last AND C_W_ID = :w_id AND C_D_ID = :d_id
  ORDER BY C_FIRST FOR FETCH ONLY


Access Table : Remote Access
  Server: SYBASE1

Remote SQL Statement:
  SELECT AA1.c_id, AA1.c_d_id, AA1.c_w_id, AA1.c_first, AA1.c_middle,
          AA1.c_last, AA1.c_balance
  FROM iwdj01.tpcc_customer AA1
  WHERE (AA1.c_w_id = :H0 ) AND (AA1.c_d_id = :H1 ) AND (AA1.c_last =
          :H2 )


==  Remote Plan from Server SYBASE1 ======================================

Residual Predicate(s)
  #Predicates = 1
  Create/Insert Into Sorted Temp Table  ID = t1
    Sort  #Columns = 1
    Piped
Sorted Temp Table Completion  ID = t1
Access Temp Table  ID = t1  #Columns = 5
  Scan Direction  = Forward
  Relation Scan
```

The report shows the access strategy for the SELECT statement on the first section of the package. In this case, the CUSTOMER table is in a Sybase data source, and the report shows the SQL statement that is sent to the data source. It also shows the access strategy for the Sybase data source.

To interpret the access strategy for your environment, refer to *IBM DataJoiner Administration*.

Using Explain for Dynamic SQL Statements: DataJoiner provides a script that can be used to explain an SQL statement entered on the command line. The Explain tool supports this function by:

1. Building a small C program that contains the statements submitted
2. Connecting to the database specified
3. Preparing the C program and creating a package
4. Running the Explain tool against the resulting package
5. Discarding the C program and object.

To invoke the Explain tool in dynamic mode, issue the following command:

```
$ dynexpln dbname "SQL statement"
```

where dbname is the name of the database in which you want to run the SQL statement.

For example, assume that we want to get the explanation for the following SQL statement:

```
SELECT T1.EMPLOYEENO, T1.NAME, T2.ANNUALSALARY
FROM IWDJ02.TS_PERSONNEL_DATA T1, IWDJ02.P3_PERSONNEL_SAL T2
WHERE (T1.EMPLOYEENO=T2.EMPLOYEENO) AND (T1.DIVISION = 'Head
        Office' )
```

The identifier IWDJ02.TS_PERSONNEL_DATA is a nickname defined in DataJoiner that points to a table stored in a Sybase data source. This table has columns named EMPLOYEENO, NAME, and DIVISION.

The identifier IWDJ02.P3_PERSONNEL_SAL, is a nickname defined in DataJoiner that points to a table stored in a DB2/MVS V3 data source. This table has columns named EMPLOYEENO and SALARY.

To get the explanation for the example SQL statement, we issue the following command:

```
$ dynexpln tpcc \
> "SELECT T1.EMPLOYEENO, \
>         T1.NAME, \
>         T2.ANNUALSALARY \
> FROM   IWDJ02.TS_PERSONNEL_DATA T1, \
>        IWDJ02.P3_PERSONNEL_SAL T2 \
> WHERE  (T1.EMPLOYEENO=T2.EMPLOYEENO) \
>        AND (T1.DIVISION =  'Head Office' )"
```

and the following report is returned:

```
DB2/6000 Version 1.1.0, 5622-044 (c) Copyright IBM Corp. 1991, 1993
Licensed Material - Program Property of IBM
IBM DATABASE 2 AIX SQL Explain Function

******************** PACKAGE ************************************

Package Name = IWDJ02.DYNEXPLN
     Prep Date = 1995/05/01
     Prep Time = 17:06:57:013

------------------- SECTION ---------------------------------------
Section = 1


SQL Statement:
  SELECT T1.EMPLOYEENO, T1.NAME, T2.ANNUALSALARY
  FROM IWDJ02.TS_PERSONNEL_DATA T1, IWDJ02.P3_PERSONNEL_SAL T2
  WHERE (T1.EMPLOYEENO=T2.EMPLOYEENO) AND (T1.DIVISION = 'Head
          Office' )


Access Table : Remote Access
  Server: SYBASE1
```

```
Remote SQL Statement:
  SELECT AA1.EMPLOYEENO, AA1.NAME, AA1.DIVISION
  FROM iwdjO2.PERSONNEL_DATA AA1
  WHERE (AA1.DIVISION = 'Head Office')


==  Remote Plan from Server SYBASE1 ======================================

Nested Loop Join
  Access Table : Remote Access
  Server: DB23A

Remote SQL Statement:
  SELECT AB1."EMPLOYEENO", AB1."ANNUALSALARY"
  FROM IWDJO2."PERSONNEL_SAL" AB1
  WHERE :HO = AB1."EMPLOYEENO"
```

The report shows the following information:

- The order in which the data sources are to be accessed
- The SQL statements that are sent to the data sources
- The type of join that DataJoiner uses to satisfy this request. In this example the type of join is a nested-loop join.

To interpret the access strategy for your SQL statements, refer to the *IBM DataJoiner Adminstration Guide*.

8.2.3 Using the DataJoiner Monitor for Remote Data Sources

The database system monitor provides a wide variety of statistical information about the operation of the database manager. This information may be controlled and accessed by coding programs that call the APIs provided or by using the following command line processor (CLP) commands:

`Get monitor switches`

Displays the current settings for the database system monitor recording switches: SORT, LOCK, TABLE, BUFFERPOOL, UOW, and STATEMENT. You must be logged in as SYSADM to issue this command.

`Update monitor switches`

Allows you to turn one or more database monitor recording switches ON or OFF. The six switches are SORT, LOCK, TABLE, BUFFERPOOL, UOW, and STATEMENT. By default they are OFF. This command can be executed only from server nodes. You must have SYSADM authority. The switches stay set until you issue a db2stop command. Each instance has its own switches.

`Reset monitor`

Sets to zero the internal database system monitor data areas of a specified database or all databases.

`Get snapshot`

Calls the database system monitor snapshot API and provides statistics on the database manager at the time of the call. You must have SYSADM authority.

`List applications`

Displays the application program name, authorization ID, agent ID, application ID, and database name. The DataJoiner-connected clients appear in this list.

`List DCS applications`

Allows you to view information about the clients connected to DDCS/6000.

`Force applications`

Terminates a client session with DataJoiner.

A complete description of this command can be found in *DB2/6000 Command Reference*.

The monitor is based on the DB2/6000 V1 code, but it has been enhanced to report statistics for the data sources that DataJoiner accesses. The commands to request statistical information from the data sources are:

- `GET SNAPSHOT FOR REMOTE_DATABASES ON dbname`
- `GET SNAPSHOT FOR REMOTE_APPLICATIONS ON dbname`

where `dbname` is the name of the DataJoiner database in the Database Directory. These commands are described in the following subsections.

8.2.3.1 Statistics for Remote Databases

The REMOTE_DATABASES option of the `GET SNAPSHOT` command reports information about the activity on the remote data source, such as type and quantity of the SQL statements, the number of rows sent or received, and the time spent in the remote data source to process the different types of SQL statements.

For example, we have two tables, TS_PERSONNEL_DATA and P3_PERSONNEL_SAL. The table TS_PERSONNEL_DATA is stored in a Sybase data source identified with the qualifier `SYBASE1` in the SYSIBM.SYSSERVERS table, and has columns named EMPLOYEENO, NAME, and DIVISION. The table P3_PERSONNEL_SAL is stored in a DB2/MVS V3 data source identified with the qualifier `DB23A` in the SYSIBM.SYSSERVERS table, and has columns named EMPLOYEENO and ANNUALSALARY.

Connecting to the database called `tpcc`, we reset the monitor switches with the command

```
reset monitor all
```

We then issue the following SQL SELECT statement which joins the two tables:

```
SELECT DECIMAL(T1."EMPLOYEENO"),
       T1."NAME",
       DECIMAL(T2."ANNUALSALARY")
FROM IWDJO2."TS_PERSONNEL_DATA" T1,
     IWDJO2."P3_PERSONNEL_SAL" T2
WHERE (T1."EMPLOYEENO"=T2."EMPLOYEENO")
  AND (T1."DIVISION" =  'Head Office' )
ORDER BY 1
```

The result is as follows (note that these tables are populated with fictitious data and do not intend to represent attributes of actual employees of any organization):

Query Results

```
1                NAME                 3
---------------- -------------------- ----------------
          100871. Bacchus, Judy                 18424.
          104188. Bailey, Joe                  173612.
          138121. Fabrizi, Gina                178218.
          143996. Auerbach, Erica               42871.
          149119. Smith, Fiona                  13109.
          151616. Reeves, Karen                110545.
          166101. Carter, Alice                 21258.
          181583. McAllister, Jane              17006.
          201420. Castro, Ian                   44643.
          244209. Mitchell, Peter               41454.
          245872. Thorpe, Sandra                17715.
          277506. Wakeland, Mickey             181761.
          326877. Goodwin, Susan                21967.
          379536. Fisher, Ruth                  18778.
          381879. Birch, Norman                 36139.
          382744. Holmes, Linda                 23030.
          425678. Austin, Liz                   17715.
          429389. Khan, Hashim                  43934.
          431027. Bates, Judy                   19841.
          432660. Biro, Karin                  165817.
          445740. Beasley, Elizabeth            21967.
          456620. Clark, Tom                    38974.
          537546. Tarrega, Frank                14526.
          547685. Warner, Ann                   18424.
          604385. Walker, Valerie              133575.
          694412. Novak, Becky                  19132.
          707114. Baxter, Mary                  47123.
          790825. Waters, Julia                 17006.
          799739. Barnett, Joseph              205854.
          815130. Hunter, Edna                 116214.

  30 record(s) selected.
```

Before disconnecting from the tpcc database, we issue the following command to determine the activity in the data sources:

```
get snapshot for remote_databases on tpcc
```

This command produces the following reports:

Activity on the DB2/MVS V3 Data Source

```
              Remote Database Snapshot

Remote datasource name                     = DB23A
Database name                              = TPCC

Connects                                   = 1
Disconnects                                = 0
Commits                                    = 1
Rollbacks                                  = 0
Queries                                    = 30
Inserts                                    = 0
Updates                                    = 0
Deletes                                    = 0
Create nicknames                           = 0
Passthrus                                  = 0

Rows returned                              = 30
Rows updated                               = 0
Rows deleted                               = 0
Rows inserted                              = 0
Failed statements                          = 0

Query time (milliseconds)                  = 21081
Insert time (milliseconds)                 = 0
Update time (milliseconds)                 = 0
Delete time (milliseconds)                 = 0
Create nickname time (milliseconds)        = 0
Passthru time (milliseconds)               = 0
```

Activity on the Sybase Data Source

```
              Remote Database Snapshot

Remote datasource name                     = SYBASE1
Database name                              = TPCC

Connects                                   = 1
Disconnects                                = 0
Commits                                    = 1
Rollbacks                                  = 0
Queries                                    = 1
Inserts                                    = 0
Updates                                    = 0
Deletes                                    = 0
Create nicknames                           = 0
Passthrus                                  = 0

Rows returned                              = 30
Rows updated                               = 0
Rows deleted                               = 0
Rows inserted                              = 0
Failed statements                          = 0

Query time (milliseconds)                  = 1239
Insert time (milliseconds)                 = 0
Update time (milliseconds)                 = 0
Delete time (milliseconds)                 = 0
Create nickname time (milliseconds)        = 0
Passthru time (milliseconds)               = 0
```

From the reports, we can see the number of queries sent to each data source, the number of rows they return, and the time spent to satisfy the request. You must not disconnect from the database before running the GET SNAPSHOT command, because if you disconnect from the database first, the monitor loses the information.

If you want to see the actual access strategy, use the Explain tool as described in 8.2.2, “Using the DataJoiner Explain Tool” on page 181.

If you want to get the data sources statistics for all the databases in the DataJoiner instance, you must issue the following command:

```
get snapshot for all remote_databases
```

8.2.3.2 Statistics for Remote Applications

The REMOTE_APPLICATIONS option of the GET SNAPSHOT command reports, for every application agent ID, information about the activity on the remote data sources such as type and quantity of the SQL statements, the number of rows sent or received, and the time spent in the remote data source to process the different types of SQL statements.

To get the report for remote applications you must issue the following command from the CLP:

```
get snapshot for remote_applications on dbname
```

If you want the report for all the remote applications on all the databases of the DataJoiner instance, you must issue the following command:

```
get snapshot for all remote_applications
```

This command, however, shows the statistics for only those applications that are connected during the execution of the command.

Appendix A. AIX Definitions in Germany

This appendix contains definitions made in the AIX system in Germany, where DataJoiner was installed.

SNA/6000 Profile - part 1

```
sna:
 prof_name                                        = "sna"
 max_sessions                                     = 200
 max_conversations                                = 200
 restart_action                                   = once
 rrm_enabled                                      = no
 dynamic_inbound_partner_lu_definitions_allowed = yes
 standard_output_device                           = "/dev/console"
 standard_error_device                            = "/var/sna/sna.stderr"
 nmvt_action_when_no_nmvt_process                 = reject
 comments                                         = ""

control_pt:
 prof_name                                        = "node_cp"
 xid_node_id                                      = "*"
 network_name                                     = "DEIBMIPF"
 control_pt_name_alias                            = "IPFP221B"
 control_pt_name                                  = "IPFP221B"
 control_pt_node_type                             = appn_end_node
 max_cached_trees                                 = 500
 max_nodes_in_topology_database                   = 500
 route_addition_resistance                        = 128
 comments                                         = ""

local_lu_lu6.2:
 prof_name                                        = "ddcslu01"
 local_lu_name                                    = "IPFBOEDJ"
 local_lu_alias                                   = "IPFBOEDJ"
 local_lu_dependent                               = no
 local_lu_address                                 =
 sscp_id                                          = *
 link_station_prof_name                           = ""
 conversation_security_list_profile_name          = ""
 comments                                         = ""

partner_lu6.2_location:
 prof_name                                        = "ddcsdbvm"
 fq_partner_lu_name                               = "DEIBMIPF.IPFA2GL4"
 partner_location_method                          = link_station
 fq_partner_owning_cp_name                        = ""
 local_node_is_network_server_for_len_node        = no
 fq_node_server_name                              = ""
 local_lu_name                                    = "IPFBOEDJ"
 link_station_profile_name                        = "ddcs3174"
 comments                                         = ""
```

© Copyright IBM Corp. 1995

SNA/6000 Profile - part 2

```
partner_lu6.2_location:
 prof_name                                  = "pokdb2"
 fq_partner_lu_name                         = "USIBMSC.SCLUDB3A"
 partner_location_method                    = link_station
 fq_partner_owning_cp_name                  = ""
 local_node_is_network_server_for_len_node  = no
 fq_node_server_name                        = ""
 local_lu_name                              = "IPFBOEDJ"
 link_station_profile_name                  = "ddcs3174"
 comments                                   = ""

partner_lu6.2_location:
 prof_name                                  = "VSEESA"
 fq_partner_lu_name                         = "DEIBMIPF.IPFA21CD"
 partner_location_method                    = link_station
 fq_partner_owning_cp_name                  = ""
 local_node_is_network_server_for_len_node  = no
 fq_node_server_name                        = ""
 local_lu_name                              = "IPFBOEDJ"
 link_station_profile_name                  = "ddcs3174"
 comments                                   = ""

partner_lu6.2_location:
 prof_name                                  = "DBSRV4"
 fq_partner_lu_name                         = "DEIBMIPF.IPFCLOEO"
 partner_location_method                    = link_station
 fq_partner_owning_cp_name                  = ""
 local_node_is_network_server_for_len_node  = no
 fq_node_server_name                        = ""
 local_lu_name                              = "IPFBOEDJ"
 link_station_profile_name                  = "DBSRV4"
 comments                                   = ""

side_info:
 prof_name                                  = "boevmis2"
 local_lu_or_control_pt_alias               = "IPFBOEDJ"
 partner_lu_alias                           = ""
 fq_partner_lu_name                         = "DEIBMIPF.IPFA2GL4"
 mode_name                                  = "IBMRDBM"
 remote_tp_name_in_hex                      = no
 remote_tp_name                             = "S34VMDBO"
 comments                                   = ""

side_info:
 prof_name                                  = "sdpokdb2"
 local_lu_or_control_pt_alias               = "IPFBOEDJ"
 partner_lu_alias                           = ""
 fq_partner_lu_name                         = "USIBMSC.SCLUDB3A"
 mode_name                                  = "IBMRDBM"
 remote_tp_name_in_hex                      = yes
 remote_tp_name                             = "07F6C4C2"
 comments                                   = ""
```

SNA/6000 Profile - part 3

```
side_info:
 prof_name                                  = "DBSRV4"
 local_lu_or_control_pt_alias               = "IPFBOEDJ"
 partner_lu_alias                           = ""
 fq_partner_lu_name                         = "DEIBMIPF.IPFCLOE0"
 mode_name                                  = "IBMRDBM"
 remote_tp_name_in_hex                      = no
 remote_tp_name                             = "DB2DUMMY"
 comments                                   = "DB2/2 on DBSRV4"

side_info:
 prof_name                                  = "DBVSETPN"
 local_lu_or_control_pt_alias               = "IPFBOEDJ"
 partner_lu_alias                           = ""
 fq_partner_lu_name                         = "DEIBMIPF.IPFA21CD"
 mode_name                                  = "IBMRDBM"
 remote_tp_name_in_hex                      = no
 remote_tp_name                             = "TPN1"
 comments                                   = ""

local_tp:
 prof_name                                  = "OS2TP"
 tp_name                                    = "OS2TP"
 tp_name_in_hex                             = no
 pip_data_present                           = no
 pip_data_subfields_number                  = 0
 conversation_type                          = basic
 sync_level                                 = none/confirm
 resource_security_level                    = none
 resource_access_list_profile_name          = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2acntp"
 multiple_instances                         = yes
 user_id                                    = 1000
 server_synonym_name                        = ""
 restart_action                             = once
 communication_type                         = signals
 ipc_queue_key                              = 0
 standard_input_device                      = "/dev/console"
 standard_output_device                     = "/dev/console"
 standard_error_device                      = "/dev/console"
 comments                                   = ""

local_tp:
 prof_name                                  = "OS2INT"
 tp_name                                    = "OS2INT"
 tp_name_in_hex                             = no
 pip_data_present                           = no
 pip_data_subfields_number                  = 0
 conversation_type                          = either
 sync_level                                 = none/confirm
 resource_security_level                    = none
 resource_access_list_profile_name          = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2cnsm"
 multiple_instances                         = yes
 user_id                                    = 1000
 server_synonym_name                        = ""
 restart_action                             = once
```

SNA/6000 Profile - part 4

```
 communication_type                          = signals
 ipc_queue_key                               = 0
 standard_input_device                       = "/dev/console"
 standard_output_device                      = "/dev/console"
 standard_error_device                       = "/dev/console"
 comments                                    = ""

local_tp:
 prof_name                                   = "zzservertp"
 tp_name                                     = "zzservertp"
 tp_name_in_hex                              = no
 pip_data_present                            = no
 pip_data_subfields_number                   = 0
 conversation_type                           = basic
 sync_level                                  = none/confirm
 resource_security_level                     = none
 resource_access_list_profile_name           = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2acntp"
 multiple_instances                          = yes
 user_id                                     = 1000
 server_synonym_name                         = ""
 restart_action                              = once
 communication_type                          = signals
 ipc_queue_key                               = 0
 standard_input_device                       = "/dev/console"
 standard_output_device                      = "/dev/console"
 standard_error_device                       = "/dev/console"
 comments                                    = ""

local_tp:
 prof_name                                   = "zzserverint"
 tp_name                                     = "DB2INTERRUPT"
 tp_name_in_hex                              = no
 pip_data_present                            = no
 pip_data_subfields_number                   = 0
 conversation_type                           = basic
 sync_level                                  = none/confirm
 resource_security_level                     = none
 resource_access_list_profile_name           = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2alttp"
 multiple_instances                          = yes
 user_id                                     = 1000
 server_synonym_name                         = ""
 restart_action                              = once
 communication_type                          = signals
 ipc_queue_key                               = 0
 standard_input_device                       = "/dev/console"
 standard_output_device                      = "/dev/console"
 standard_error_device                       = "/dev/console"
 comments                                    = ""
```

SNA/6000 Profile - part 5

```
local_tp:
 prof_name                                  = "zzserveros2tp"
 tp_name                                    = "07F6C4C2"
 tp_name_in_hex                             = yes
 pip_data_present                           = no
 pip_data_subfields_number                  = 0
 conversation_type                          = basic
 sync_level                                 = none/confirm
 resource_security_level                    = none
 resource_access_list_profile_name          = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2acntp"
 multiple_instances                         = yes
 user_id                                    = 1000
 server_synonym_name                        = ""
 restart_action                             = once
 communication_type                         = signals
 ipc_queue_key                              = 0
 standard_input_device                      = "/dev/console"
 standard_output_device                     = "/dev/console"
 standard_error_device                      = "/dev/console"
 comments                                   = ""

local_tp:
 prof_name                                  = "zzserveros2int"
 tp_name                                    = "07F6E2D5"
 tp_name_in_hex                             = yes
 pip_data_present                           = no
 pip_data_subfields_number                  = 0
 conversation_type                          = basic
 sync_level                                 = none/confirm
 resource_security_level                    = none
 resource_access_list_profile_name          = ""
 full_path_tp_exe = "/home/djinst1/sqllib/bin/db2cnsm"
 multiple_instances                         = yes
 user_id                                    = 1000
 server_synonym_name                        = ""
 restart_action                             = once
 communication_type                         = signals
 ipc_queue_key                              = 0
 standard_input_device                      = "/dev/console"
 standard_output_device                     = "/dev/console"
 standard_error_device                      = "/dev/console"
 comments                                   = ""

link_station_token_ring:
 prof_name                                  = "ddcs3174"
 use_control_pt_xid                         = no
 xid_node_id                                = 0x071e001b
 sna_dlc_profile_name                       = "tok0.00001"
 stop_on_inactivity                         = no
 time_out_value                             = 0
 LU_registration_supported                  = no
 LU_registration_profile_name               = ""
 link_tracing                               = no
 trace_format                               = long
 access_routing_type                        = link_address
```

SNA/6000 Profile - part 6

```
 remote_link_name                          = ""
 remote_link_address                       = 0x400020201001
 remote_sap                                = 0x04
 verify_adjacent_node                      = no
 net_id_of_adjacent_node                   = "DEIBMIPF"
 cp_name_of_adjacent_node                  = ""
 xid_node_id_of_adjacent_node              = "*"
 node_type_of_adjacent_node                = learn
 solicit_sscp_sessions                     = yes
 call_out_on_activation                    = yes
 activate_link_during_system_init          = yes
 activate_link_on_demand                   = no
 cp_cp_sessions_supported                  = yes
 cp_cp_session_support_required            = no
 adjacent_node_is_preferred_server         = no
 initial_tg_number                         = 0
 restart_on_normal_deactivation            = no
 restart_on_abnormal_deactivation          = no
 restart_on_activation                     = no
 TG_effective_capacity                     = 15974400
 TG_connect_cost_per_time                  = 0
 TG_cost_per_byte                          = 0
 TG_security                               = nonsecure
 TG_propagation_delay                      = lan
 TG_user_defined_1                         = 128
 TG_user_defined_2                         = 128
 TG_user_defined_3                         = 128
 comments                                  = ""

link_station_token_ring:
 prof_name                                 = "DBSRV4"
 use_control_pt_xid                        = no
 xid_node_id                               = 0x071e001b
 sna_dlc_profile_name                      = "tok0.00001"
 stop_on_inactivity                        = no
 time_out_value                            = 0
 LU_registration_supported                 = no
 LU_registration_profile_name              = ""
 link_tracing                              = no
 trace_format                              = long
 access_routing_type                       = link_address
 remote_link_name                          = ""
 remote_link_address                       = 0x40001010100e
 remote_sap                                = 0x04
 verify_adjacent_node                      = no
 net_id_of_adjacent_node                   = "DEIBMIPF"
 cp_name_of_adjacent_node                  = ""
 xid_node_id_of_adjacent_node              = "*"
 node_type_of_adjacent_node                = learn
 solicit_sscp_sessions                     = yes
 call_out_on_activation                    = yes
 activate_link_during_system_init          = yes
 activate_link_on_demand                   = no
```

SNA/6000 Profile - part 7

```
 cp_cp_sessions_supported                      = yes
 cp_cp_session_support_required                = no
 adjacent_node_is_preferred_server             = no
 initial_tg_number                             = 0
 restart_on_normal_deactivation                = no
 restart_on_abnormal_deactivation              = no
 restart_on_activation                         = no
 TG_effective_capacity                         = 15974400
 TG_connect_cost_per_time                      = 0
 TG_cost_per_byte                              = 0
 TG_security                                   = nonsecure
 TG_propagation_delay                          = lan
 TG_user_defined_1                             = 128
 TG_user_defined_2                             = 128
 TG_user_defined_3                             = 128
 comments                                      = ""

sna_dlc_token_ring:
 prof_name                                     = "tok0.00001"
 datalink_device_name                          = "tok0"
 force_timeout                                 = 120
 user_defined_max_i_field                      = no
 max_i_field_length                            = 30729
 max_active_link_stations                      = 100
 num_reserved_inbound_activation               = 0
 num_reserved_outbound_activation              = 0
 transmit_window_count                         = 16
 dynamic_window_increment                      = 1
 retransmit_count                              = 8
 receive_window_count                          = 8
 priority                                      = 0
 inact_timeout                                 = 48
 response_timeout                              = 4
 acknowledgement_timeout                       = 1
 link_name                                     = ""
 local_sap                                     = 0x04
 retry_interval                                = 60
 retry_limit                                   = 20
 dynamic_link_station_supported                = yes
 trace_base_listen_link_station                = no
 trace_base_listen_link_station_format         = long
 dynamic_lnk_solicit_sscp_sessions             = yes
 dynamic_lnk_cp_cp_sessions_supported          = yes
 dynamic_lnk_cp_cp_session_support_required    = no
 dynamic_lnk_TG_effective_capacity             = 15974400
 dynamic_lnk_TG_connect_cost_per_time          = 0
 dynamic_lnk_TG_cost_per_byte                  = 0
 dynamic_lnk_TG_security                       = nonsecure
 dynamic_lnk_TG_propagation_delay              = lan
 dynamic_lnk_TG_user_defined_1                 = 128
 dynamic_lnk_TG_user_defined_2                 = 128
 dynamic_lnk_TG_user_defined_3                 = 128
 comments                                      = ""
```

SNA/6000 Profile - part 8

```
mode:
 prof_name                                = "IBMRDB"
 mode_name                                = "IBMRDB"
 max_sessions                             = 20
 min_conwinner_sessions                   = 10
 min_conloser_sessions                    = 10
 auto_activate_limit                      = 0
 max_adaptive_receive_pacing_window       = 16
 receive_pacing_window                    = 7
 max_ru_size                              = 1920
 min_ru_size                              = 256
 class_of_service_name                    = "#CONNECT"
 comments                                 = ""

mode:
 prof_name                                = "IBMRDBM"
 mode_name                                = "IBMRDBM"
 max_sessions                             = 10
 min_conwinner_sessions                   = 5
 min_conloser_sessions                    = 5
 auto_activate_limit                      = 0
 max_adaptive_receive_pacing_window       = 16
 receive_pacing_window                    = 3
 max_ru_size                              = 2816
 min_ru_size                              = 1024
 class_of_service_name                    = "#CONNECT"
 comments                                 = ""

conv_list:
 prof_name                                = "datajoiner"
 username_list                            = {mmres1,mmres2,mmres3,mmres5}
 comments                                 = ""
```

Database Manager Configuration

```
Database manager configuration release level  = 0x0500
Node type                                      = Server with remote client

Service name                                (SVCENAME) = djxtcpip
Transaction program name                      (TPNAME) = zzservertp

Max requester I/O block size (bytes)         (RQRIOBLK) = 4096
Max server I/O block size (bytes)            (SVRIOBLK) = 4096
Communication heap size (4KB)               (COMHEAPSZ) = 128
Remote services heap size (4KB)              (RSHEAPSZ) = 128
Sort heap threshold (4KB)                   (SHEAPTHRES) = 4096
Application support layer heap size (4KB)    (ASLHEAPSZ) = 100

Max no. of existing agents                   (MAXAGENTS) = 200
Max no. of concurrent agents                (MAXCAGENTS) = MAXAGENTS
Max no. of concurrently active databases         (NUMDB) = 8

Application cleanup interval (ms)           (CUINTERVAL) = 5000
Keep DARI process                             (KEEPDARI) = YES
Max. no. of DARI processes                     (MAXDARI) = MAXAGENTS
Priority of agents                            (AGENTPRI) = SYSTEM
Database monitor SQL statement size (bytes) (SQLSTMTSZ) = 256
Index re-creation time                         (INDEXREC) = RESTART
Default database path                         (DFTDBPATH) = /home/djinst1

Backup buffer default size (4KB)             (BACKBUFSZ) = 1024
Restore buffer default size (4KB)            (RESTBUFSZ) = 1024
```

Node Directory - part 1

```
 Node Directory

 Number of entries in the directory = 8

Node 1 entry:

 Node name                      = AIXBOE
 Comment                        = AIX System in Germany
 Protocol                       = TCPIP
 Hostname                       = loopback.pr.boeblingen.ibm.com
 Service name                   = db2tcpip

Node 2 entry:

 Node name                      = AIXDB2
 Comment                        = Second AIX system in Germany
 Protocol                       = TCPIP
 Hostname                       = itso2.pr.boeblingen.ibm.com
 Service name                   = db2tcpipl

Node 3 entry:

 Node name                      = AIXSJC
 Comment                        = AIX System in San Jose, CA
 Protocol                       = TCPIP
 Hostname                       = severn.sanjose.ibm.com
 Service name                   = severndj

Node 4 entry:

 Node name                      = DBSRV4
 Comment                        = DB2/2 V2 under OS/2 in Germany
 Protocol                       = CPIC
 Symbolic destination name      = DBSRV4
 Security type                  = PROGRAM

Node 5 entry:

 Node name                      = DJDB104
 Comment                        = Second DJ Database (same instance)
 Protocol                       = TCPIP
 Hostname                       = loopback.pr.boeblingen.ibm.com
 Service name                   = djxtcpip

Node 6 entry:

 Node name                      = MVSESA
 Comment                        = MVS System in the USA
 Protocol                       = CPIC
 Symbolic destination name      = sdpokdb2
 Security type                  = PROGRAM
```

Node Directory - part 2

```
Node 7 entry:

 Node name                      = VMESA
 Comment                        = BOEVMIS2 System in Germany
 Protocol                       = CPIC
 Symbolic destination name      = boevmis2
 Security type                  = PROGRAM

Node 8 entry:

 Node name                      = VSEESA
 Comment                        = VSEANL13 System in Germany
 Protocol                       = CPIC
 Symbolic destination name      = DBVSETPN
 Security type                  = PROGRAM
```

System Database Directory - part 1

```
 Number of entries in the directory = 10

Database 1 entry:

 Database alias                 = DJREL104
 Database name                  = DJREL104
 Local database directory       = /home/djinst1
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        = DataJoiner Database in Germany
 Directory entry type           = Indirect
 Authentication                 = SERVER

Database 2 entry:

 Database alias                 = DB2VSE
 Database name                  = S34VSDB1
 Local database directory       =
 Database directory             =
 Node name                      = VSEESA
 Database release level         = 5.00
 Comment                        = VSE SQL/DS in Germany
 Directory entry type           = Remote
 Authentication                 = DCS

Database 3 entry:

 Database alias                 = DB2VM
 Database name                  = S34VMDBO
 Local database directory       =
 Database directory             =
 Node name                      = VMESA
 Database release level         = 5.00
 Comment                        = VM SQL/DS in Germany
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 4 entry:

 Database alias                 = DBSJCDJ
 Database name                  = DBINST1
 Local database directory       =
 Database directory             =
 Node name                      = AIXSJC
 Database release level         = 5.00
 Comment                        = DataJoiner in San Jose, CA
 Directory entry type           = Remote
 Authentication                 = DCS
```

System Database Directory - part 2

```
Database 5 entry:

 Database alias                 = DBSRV4D
 Database name                  = DBSRV4D
 Local database directory       =
 Database directory             =
 Node name                      = DBSRV4
 Database release level         = 5.00
 Comment                        = DB2/2 Database in Germany
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 6 entry:

 Database alias                 = DB2MVS
 Database name                  = CENTSJC
 Local database directory       =
 Database directory             =
 Node name                      = MVSESA
 Database release level         = 5.00
 Comment                        = DB2/MVS in the USA
 Directory entry type           = Remote
 Authentication                 = DCS

Database 7 entry:

 Database alias                 = DB26KREM
 Database name                  = SAMPLE
 Local database directory       =
 Database directory             =
 Node name                      = AIXDB2
 Database release level         = 5.00
 Comment                        = DB2/6000 in AIXDB2 System
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 8 entry:

 Database alias                 = DATAJOIN
 Database name                  = DATAJOIN
 Local database directory       = /home/djinst1
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        = DataJoiner Database in Germany
 Directory entry type           = Indirect
 Authentication                 = SERVER
```

System Database Directory - part 3

```
Database 9 entry:

 Database alias                 = DB26000
 Database name                  = BOE6000
 Local database directory       =
 Database directory             =
 Node name                      = AIXBOE
 Database release level         = 5.00
 Comment                        = DB2/6000 Database in Germany
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 10 entry:

 Database alias                 = DJDB104
 Database name                  = DJREL104
 Local database directory       =
 Database directory             =
 Node name                      = DJDB104
 Database release level         = 5.00
 Comment                        = 2nd DJ DB, defined as remote DB
 Directory entry type           = Remote
 Authentication                 = SERVER
```

SYSIBM.SYSSERVERS

SERVER	NODE	DBNAME	TYPE	VERSION	PROTOCOL	PASSWORD
DB26KLOC	AIXBOE	BOE6000	DB2/6000	1.1	jra	Y
DB26KREM	AIXDB2	SAMPLE	DB2/6000	1.1	jra	Y
DB26KTST	AIXBOE	MARCIO	DB2/6000	1.1	jra	Y
DBSJCDJ	AIXSJC	DBINST1	DATAJOINER	1.1	jra	Y
DB2VM	VMESA	S34VMDBO	DB2/VM	3.4	drda	Y
DB2VSE	VSEESA	S34VSDB1	DB2/VSE	3.4	drda	Y
DB2MVS	MVSESA	CENTSJC	DB2/MVS	3.1	drda	Y
DJLOOP	DJDB104	DJREL104	DATAJOINER	1.1	jra	Y
DBSRV4D	DBSRV4	DBSRV4D	DB2/2	2.1	jra	Y

SYSIBM.SYSREMOTEUSERS

AUTHID	SERVER	REMOTE_AUTHID	REMOTE_PW
BOEUS1	DB26KLOC	boeus1	*
BOEUS1	DB26KREM	boeus1	*
BOEUS1	DB26KTST	boeus1	*
BOEUS1	DBSJCDJ	mmres1	*
BOEUS1	DB2VSE	BOEUS1	*
BOEUS1	DB2VM	BOEUS1	*
BOEUS1	DB2MVS	BOEUS1	*
BOEUS1	DBSRV4D	BOEUS1	*
DJUSER1	DB2VM	BOEUS2	*
DJUSER1	DB2VSE	BOEUS2	*

10 record(s) selected

Appendix B. VSE/ESA Definitions in Germany

This appendix contains definitions made in the VSE/ESA system in Germany in order to provide DRDA access to VSE SQL/DS with DataJoiner.

VSE/VTAM Parameters

```
SSCPID=21,
HOSTSA=21,
SSCPNAME=IPFV2A,
HOSTPU=IPFVM21,
NOPROMPT,
NETID=DEIBMIPF,
MAXSUBA=255,
CONFIG=00,
DYNLU=YES,
IOINT=0,
SGALIMIT=0,
BSBUF=(28,,,1),
CRPLBUF=(60,,,1),
LFBUF=(70,288,,11),
LPBUF=(12,,,6),
SFBUF=(20,,,20),
SPBUF=(210,,,32),
VFBUF=102400,
VPBUF=446464,
XDBUF=(6,,,1)
```

© Copyright IBM Corp. 1995

CICS Signon Table (DFHSNT)

```
         TITLE 'DFHTSNT -- FOR DRDA REQUEST'
         PUNCH ' CATALOG DFHSNT.OBJ REP=YES'
         DFHSNT TYPE=INITIAL,EXTSEC=NO
*************************************************
* **** VM DRDA USERS ****************************
*************************************************
* **** CMS USERS VMSQLUSO TO VMSQLUSA ***************
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=VMSQLUSO,PASSWRD=PWDRDA
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=VMSQLUS1,PASSWRD=PWDRDA
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=VMSQLUS2,PASSWRD=PWDRDA
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=VMSQLUS3,PASSWRD=PWDRDA
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=VMSQLUS3,PASSWRD=PWDRDA
*************************************************
* **** OS/2  DRDA USERS *****************************
*************************************************
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=NULLID,PASSWRD=NULLID
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DBSRV1,PASSWRD=SSMC1PW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DBSRV2,PASSWRD=SSMC1PW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DBSRV3,PASSWRD=SSMC1PW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DBREQ1,PASSWRD=SSMC1PW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DBREQ2,PASSWRD=SSMC1PW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DEMO,PASSWRD=SQL34PW
*****************************************************************
* OTHER DRDA USERIDS                                            **
*****************************************************************
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=ROCHUS1,PASSWRD=DRDAPW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=BOCAUS2,PASSWRD=DRDAPW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=SJUS3,PASSWRD=DRDAPW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=RALUS4,PASSWRD=DRDAPW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=BOEUS1,PASSWRD=BOEXUS
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=BOEUS2,PASSWRD=BOEXUS
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=BOEUS3,PASSWRD=BOEXUS
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DJUSER1,PASSWRD=DJXUSER
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DJUSER2,PASSWRD=DJXUSER
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=DJUSER3,PASSWRD=DJXUSER
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=AUSTUS5,PASSWRD=DRDAPW
         DFHSNT TYPE=ENTRY,RSLKEY=0,USERID=POKUS6,PASSWRD=DRDAPW
         DFHSNT TYPE=FINAL
         END
/*
```

CEDA Connection Definition for DJ in Germany

```
CEDA  VIEW
 Connection     : OEDJ
 Group          : DRDA
CONNECTION IDENTIFIERS
 Netname        : IPFBOEDJ
 INDsys         :
REMOTE ATTRIBUTES
 REMOTESystem   :
 REMOTEName     :
CONNECTION PROPERTIES
 ACcessmethod   : Vtam              Vtam | IRc | INdirect
 Protocol       : Appc              Appc | Lu61
 SInglesess     : No                No | Yes
 Datastream     : User              User | 3270 | SCs | STrfield | Lms
 RECordformat   : U                 U | Vb
OPERATIONAL PROPERTIES
 AUtoconnect    : Yes               No | Yes | All
 INService      : Yes               Yes | No
SECURITY
 SEcurityname   :
 ATtachsec      : Verify            Local | Identify | Verify
 Bindpassword   :                   PASSWORD NOT SPECIFIED
```

CEDA Session Definition for DJ in Germany

```
CEDA  VIEW
 Sessions       : OEDJ
 Group          : DRDA
SESSION IDENTIFIERS
 Connection     : OEDJ
 SESSName       :
 NETnameq       :
 MOdename       : IBMRDBM
SESSION PROPERTIES
 Protocol       : Appc               Appc | Lu61
 MAximum        : 00005 , 00002      0-32767
 RECEIVEPfx     :
 RECEIVECount   : No                 No | 1-999
 SENDPfx        :
 SENDCount      : No                 No | 1-999
 SENDSize       : 04096              1-30720
 RECEIVESize    : 04096              1-30720
OPERATOR DEFAULTS
 OPERId         :
 OPERPriority   : 000                0-255
 OPERRsl        : 0
 OPERSecurity   : 1
 USERId         :
SESSION USAGES
 Transaction    :
 SESSPriority   : 000                0-255
OPERATIONAL PROPERTIES
 Autoconnect    : Yes                No | Yes | All
 INservice      :                    No | Yes
 Buildchain     : Yes                Yes | No
 USERArealen    : 000                0-255
 IOarealen      : 00000 , 00000      0-32767
 RELreq         : No                 No | Yes
 Discreq        : No                 No | Yes
 NEPclass       : 000                0-255
RECOVERY
 RECOvoption    : Sysdefault         Sysdefault | None
```

CEDA Connection Definition for DJ in San Jose

```
OBJECT CHARACTERISTICS
 CEDA  View
  Connection     : O85I
  Group          : DRDA
 CONNECTION IDENTIFIERS
  Netname        : SCA2085I
  INDsys         :
 REMOTE ATTRIBUTES
  REMOTESystem   :
  REMOTEName     :
 CONNECTION PROPERTIES
  ACcessmethod   : Vtam              Vtam | IRc | INdirect
  Protocol       : Appc              Appc | Lu61
  SInglesess     : No                No | Yes
  Datastream     : User              User | 3270 | SCs | STrfield | Lms
  RECordformat   : U                 U | Vb
 OPERATIONAL PROPERTIES
  AUtoconnect    : Yes               No | Yes | All
  INService      : Yes               Yes | No
 SECURITY
  SEcurityname   :
  ATtachsec      : Verify            Local | Identify | Verify
  Bindpassword   :                   PASSWORD NOT SPECIFIED
```

CEDA Session Definition for DJ in San Jose

```
CEDA  View
 Sessions       : 085I
 Group          : DRDA
SESSION IDENTIFIERS
 Connection     : 085I
 SESSName       :
 NETnameq       :
 MOdename       : IBMRDBM
SESSION PROPERTIES
 Protocol       : Appc              Appc | Lu61
 MAximum        : 00020 , 00010     0-32767
 RECEIVEPfx     :
 RECEIVECount   : No                No | 1-999
 SENDPfx        :
 SENDCount      : No                No | 1-999
 SENDSize       : 04096             1-30720
 RECEIVESize    : 04096             1-30720
OPERATOR DEFAULTS
 OPERId         :
 OPERPriority   : 000               0-255
 OPERRsl        : 0
 OPERSecurity   : 1
 USERId         :
SESSION USAGES
 Transaction    :
 SESSPriority   : 000               0-255
OPERATIONAL PROPERTIES
 Autoconnect    : Yes               No | Yes | All
 INservice      :                   No | Yes
 Buildchain     : Yes               Yes | No
 USERArealen    : 000               0-255
 IOarealen      : 00000 , 00000     0-32767
 RELreq         : No                No | Yes
 Discreq        : No                No | Yes
 NEPclass       : 000               0-255
RECOVERY
 RECOvoption    : Sysdefault        Sysdefault | None
```

VSE Database Directory

```
*TPN     APPLID     *DBNAME                PID   PRIV                *
*        1      1   22                3    44    5                   *
*2..5___0......7___12.............9___45___0___________________*
 TPN0    SYSARI00    S34VSDB0              F8
 TPN1    SYSARI01    S34VSDB1
 TPN2    SYSARI02    S34VSDB2
 TPN3    SYSARI03    S34VSDBP                    Y
         SYSARI02    S34VSDB2              01
         SYSARI01    S34VSDB1              0B
         SYSARI01    S34VSDB1              05
         SYSARI01    S34VSDB1              0A
         SYSARI01    S34VSDB1              06
         SYSARI01    S34VSDB1              07
         SYSARI01    S34VSDB1              08
         SYSARI01    S34VSDB1              09
         SYSARI03    S34VSDBP
         SYSARI01   *S34VSDB1
         SQLMDSK     SQLMDSK
         S34VMDB0    S34VMDB0
         S34VMDB1    S34VMDB1
         S34VMDB2    S34VMDB2
         S34VMDB3    S34VMDB3
         S34VMDB4    S34VMDB4
         S34VMDB5    S34VMDB5
         S34VMDB6    S34VMDB6
         S34VMDB7    S34VMDB7
         S34VMDB8    S34VMDB8
         S34VMDB9    S34VMDB9
         S34VMDBA    S34VMDBA
         S34VMDBB    S34VMDBB
 6DB    SYSARI00    SQLDS
```

VSE SQL/DS Startup

```
* $$ JOB JNM=S34VSDB1,CLASS=Z,DISP=L
* $$ LST CLASS=B,DISP=D,PRI=3,DEST=(*,IS2DBA)
// JOB S34VSDB1  S34VSDB1 DATABASE STARTUP IN MUM
// SETPFIX LIMIT=1024K
// TLBL ARIARCH
// LIBDEF PROC,SEARCH=(PRD2.SQL340)
// EXEC  PROC=ARIS34PL    *-- SQL/DS PRODUCTION LIBRARY ID PROC
// EXEC  PROC=S34DB1DB    *-- SQL/DS DATABASE ID PROC
// EXEC  ARISQLDS,SIZE=AUTO,PARM='DBNAME=S34VSDB1,NPAGBUF=1500,        C
               NCUSERS=15,CHKINTVL=100,NDIRBUF=2000,RMTUSERS=10'
/*
/&
* $$ EOJ
```

Appendix C. VM/ESA Definitions in Germany

This appendix contains definitions made in the VM/ESA system in Germany in order to provide DRDA access to VM SQL/DS with DataJoiner.

VM VTAM SNA Definitions

```
         VBUILD TYPE=LOCAL
*-------------------------------------------------------------------
*        DEFINE THE GATEWAY CONTROLLER PU
*-------------------------------------------------------------------
IPFCP200 PU    CUADDR=200,                                          *
               DELAY=0.2,                                           *
               ISTATUS=ACTIVE,                                      *
               MAXBFRU=29,                                          *
               PUTYPE=2,                                            *
               XID=YES,                                             *
               DYNLU=YES,                                           *
               USSTAB=ISTSNA,MODETAB=ISTINCLM,DLOGMOD=D4A32782
```

© Copyright IBM Corp. 1995

VM VTAM Cross-Domain Resources

```
         VBUILD TYPE=CDRSC
IPFA21CD CDRSC  CDRM=IPFV2A       ** SA=21 VSEANL13

IPFBOEDJ CDRSC  ALSLIST=(IPFP221B,IPFCP200),                          X
                ALSREQ=YES,                                           X
                DLOGMOD=IBMRDBM,                                      X
                MODETAB=DRDAMOD
```

VM VTAM AVSVM APPL Definition

```
IPF2AVS  VBUILD TYPE=APPL
IPFA2GL3 APPL  APPC=YES,                                          X
               AUTHEXIT=YES,                                      X
               AUTOSES=20,                                        X
               DSESLIM=200,                                       X
               DMINWNL=100,                                       X
               DMINWNR=100,                                       X
               EAS=9999,                                          X
               MAXPVT=100K,                                       X
               SECACPT=ALREADYV,                                  X
               VERIFY=NONE,                                       X
               VPACING=2,                                         X
               MODETAB=RDSTAB,                                    X
               DLOGMOD=IBMROS2,                                   X
               SYNCLVL=CONFIRM,                                   X
               OPERCNOS=ALLOW,                                    X
               PARSESS=YES
IPFA2GL4 APPL  APPC=YES,                                          X
               AUTHEXIT=YES,                                      X
               AUTOSES=20,                                        X
               DSESLIM=200,                                       X
               DMINWNL=100,                                       X
               DMINWNR=100,                                       X
               EAS=3999,                                          X
               MAXPVT=200K,                                       X
               SECACPT=ALREADYV,                                  X
               VERIFY=NONE,                                       X
               VPACING=2,                                         X
               MODETAB=DRDAMOD,                                   X
               DLOGMOD=IBMRDBM,                                   X
               SYNCLVL=CONFIRM,                                   X
               OPERCNOS=ALLOW,                                    X
               PARSESS=YES
*              STATOPT='AVSVM LU'
```

AVSVM GCS Profile

```
/*********************************************************************/
/*  The Application major node IPF2AVS defines the                   */
/*  IPFA2GLG application LU.                                         */
/*********************************************************************/
Trace O
'AGW ACTIVATE  GATEWAY IPFA2GL3 GLOBAL'
'AGW CNOS IPFA2GL3 IPFT1T1O IBMROS2 5O 25 25'
/*                                                                   */
/*********************************************************************/
/*                                                                   */
'AGW ACTIVATE  GATEWAY IPFA2GL4 GLOBAL'
'AGW CNOS IPFA2GL4 IPFA21CD IBMRDBM 2O 1O 1O'
/*                                                                   */
/* ************************************************** */
/* ** ITSO DRDA CROSS CENTER PROJECT                ** */
/* ************************************************** */
'AGW ADD USERID SCHASMO1 * ='
'AGW ADD USERID SCHASMO2 * ='
'AGW ADD USERID SCLUDB3A * ='
'AGW ADD USERID SCLUDB3B * ='
'AGW ADD USERID SJA2108I * ='
'AGW ADD USERID BOA3OOOK * ='
```

VM SQL/DS profile

```
/*********************************************************************/
/* PROFILE FOR SQL/DS SERVICE MACHINE - S34VMDBO                     */
/*********************************************************************/
Address "COMMAND"
"CP SET RUN  ON"
"ACCESS 193  V"
"ACCESS 195  Q"
"CP TERMINAL LINEND #"
"CP TERMINAL LINEDEL OFF"
"CP TERMINAL CHARDEL OFF"
"CP TERMINAL ESCAPE ¢"
"CP SET EMSG ON"
"CP SET PF10 IMM FLIST"
"CP SET PF12     RETRIEVE"
"CP SET PF22 IMM FLIST"
"CP SET PF24     RETRIEVE"
"GLOBAL TXTLIB CMSLIB CMSSAA TSOLIB VMLIB"
"SET LANGUAGE AMENG (ADD ARI USER"
/**/
"EXEC SQLSTART DBNAME(S34VMDBO) DCSSID(SQLDBA) PARM(PARMID=SQL34DSS)"
/**/
```

VM SQL/DS Startup Parameters

```
NCUSERS=10
NPAGBUF=1000
NDIRBUF=2000
NLRBS=20000
NLRBU=05000
CHKINTVL=100
TARGETWS=80
SEPINTDB=Y
DBMODE=G
PROTOCOL=AUTO
```

Appendix D. MVS VTAM Definitions in San Jose

This appendix contains definitions made in the MVS system in San Jose to provide DRDA access to DataJoiner.

VTAM Definition for DataJoiner PU and LUs

```
SJA2085  PU    ADDR=01,
               IDBLK=071,IDNUM=A2085,
               ANS=CONT,DISCNT=NO,
               IRETRY=NO,ISTATUS=ACTIVE,
               MAXDATA=265,MAXOUT=7,
               MAXPATH=1,
               PUTYPE=2,SECNET=NO,
               MODETAB=POKMODE,DLOGMOD=DYNRMT,
               USSTAB=USSRDYN,
               PACING=1,VPACING=2
*
SJA2085A LU    LOCADDR=002,LOGAPPL=SCGVAMP
SJA2085B LU    LOCADDR=003,LOGAPPL=SCGVAMP
SJA2085C LU    LOCADDR=004
SJA2085D LU    LOCADDR=005
SJA2085I LU    LOCADDR=0,DLOGMOD=LU62APPB
SJA2085J LU    LOCADDR=0,DLOGMOD=LU62APPB
SJA2085K LU    LOCADDR=0,DLOGMOD=LU62APPA
SJA2085L LU    LOCADDR=0,DLOGMOD=LU62APPA
```

Note:

In our systems, SJA and SCA definitions are interchangeable. In VTAM, our PUs and LUs are defined as SJA2085? and in SNA Server/6000 V2, they are defined as SCA2085?.

© Copyright IBM Corp. 1995

MVS VTAM Mode definition

```
IBMRDBM    MODEENT LOGMODE=IBMRDBM,     AGW  (SLU) TO AGW (PLU)
               PSNDPAC=X'00',           PRIMARY SEND PACING COUNT
               SRCVPAC=X'02',           SECONDARY RECEIVE PACING COUNT
               SSNDPAC=X'02',           SECONDARY SEND PACING COUNT
               FMPROF=X'13',            FM PROFILE 19 LU 6.2
               TSPROF=X'07',            TS PROFILE 7 LU 6.2
               PRIPROT=X'B0',           PRIMARY NAU PROTOCOL
               SECPROT=X'B0',           SECONDARY NAU PROT
               RUSIZES=X'8989',         8 X 2**9 = 4096
               PSERVIC=X'060200000000000000122F00'   SYSMSG/Q MODEL
```

MVS VTAM APPL Definition

```
SCLUDB3A  APPL ACBNAME=SCLUDB3A,  APPLID FOR DB2 V3
               APPC=YES,
               ATNLOSS=ALL,       NEW PARM WITH DB2 V310
               AUTH=(ACQ),        ACCESS TO VTAM FUNCTIONS
               AUTOSES=10,
               DMINWNL=25,
               DMINWNR=25,
               DSESLIM=50,       MAX NUMBER OF SESSIONS IN USE
               EAS=509,
               ENCR=NONE,
               MODETAB=AGWTAB,
               PARSESS=YES,
               SECACPT=ALREADYV,
               SONSCIP=NO,
               SRBEXIT=YES,      SRB PROCESSING IN EXIT ROUTINES
               SYNCLVL=SYNCPT,   NEW PARM FOR DB2 V310
               VERIFY=NONE,
               VPACING=2,
               VTAMFRR=NO
```

Appendix E. AIX Definitions in San Jose

This appendix contains definitions made in the AIX system in San Jose, where DataJoiner was installed.

SNA/6000 Profile - part 1

```
sna:
    prof_name                                       = "sna"
    max_sessions                                    = 200
    max_conversations                               = 200
    restart_action                                  = once
    rrm_enabled                                     = no
    dynamic_inbound_partner_lu_definitions_allowed = yes
    standard_output_device                          = "/dev/console"
    standard_error_device                           = "/var/sna/sna.stderr"
    nmvt_action_when_no_nmvt_process                = reject
    comments                                        = ""

control_pt:
    prof_name                                       = "node_cp"
    xid_node_id                                     = 0x071a2085
    network_name                                    = "USIBMSC"
    control_pt_name_alias                           = "SCA2085"
    control_pt_name                                 = "SCA2085"
    control_pt_node_type                            = appn_end_node
    max_cached_trees                                = 500
    max_nodes_in_topology_database                  = 500
    route_addition_resistance                       = 128
    comments                                        = ""

local_lu_lu6.2:
    prof_name                                       = "SCA2085I"
    local_lu_name                                   = "SCA2085I"
    local_lu_alias                                  = "SCA2085I"
    local_lu_dependent                              = no
    local_lu_address                                =
    sscp_id                                         = *
    link_station_prof_name                          = ""
    conversation_security_list_profile_name         = ""
    comments                                        = ""

partner_lu6.2_location:
    prof_name                                       = "DB3APLU"
    fq_partner_lu_name                              = "USIBMSC.SCLUDB3A"
    partner_location_method                         = link_station
    fq_partner_owning_cp_name                       = ""
    local_node_is_network_server_for_len_node       = no
    fq_node_server_name                             = ""
    local_lu_name                                   = "SCA2085I"
    link_station_profile_name                       = "trlink"
    comments                                        = ""
```

© Copyright IBM Corp. 1995

SNA/6000 Profile - part 2

```
partner_lu6.2_location:
    prof_name                                   = "SQLDSPLU"
    fq_partner_lu_name                          = "DEIBMIPF.IPFA2GL4"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "SQL/DS GERMANY"

partner_lu6.2_location:
    prof_name                                   = "DB23PLU"
    fq_partner_lu_name                          = "USIBMSC.LUDB23"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "DB2 V2.3 Pok"

partner_lu6.2_location:
    prof_name                                   = "DJYELPLU"
    fq_partner_lu_name                          = "USIBMSC.SCA2109I"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlnkyel"
    comments                                    = "DJ in yellow"

partner_lu6.2_location:
    prof_name                                   = "DB2VSPLU"
    fq_partner_lu_name                          = "DEIBMIPF.IPFA21CD"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "DB2/VSE GERMANY"

side_info:
    prof_name                                   = "db23atpn"
    local_lu_or_control_pt_alias                = "SCA2O85I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.SCLUDB3A"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = yes
    remote_tp_name                              = "07F6C4C2"
    comments                                    = ""
```

SNA/6000 Profile - part 3

```
side_info:
    prof_name                                   = "DB2VMTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "DEIBMIPF.IPFA2GL4"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "S34VMDBO"
    comments                                    = "SQL/DS GERMANY"

side_info:
    prof_name                                   = "db223tpn"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.LUDB23"
    mode_name                                   = "IBMRDB"
    remote_tp_name_in_hex                       = yes
    remote_tp_name                              = "07F6C4C2"
    comments                                    = "DB2 V2 Pok"

side_info:
    prof_name                                   = "DJYELTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.SCA2109I"
    mode_name                                   = "IBMRDB"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "zzservertp"
    comments                                    = "DJ in yellow"

side_info:
    prof_name                                   = "DB2VSTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "DEIBMIPF.IPFA21CD"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "TPN1"
    comments                                    = "DB2/VSE GERMANY"
```

SNA/6000 Profile - part 4

```
local_tp:
    prof_name                                   = "zzservertp"
    tp_name                                     = "zzservertp"
    tp_name_in_hex                              = no
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe     = "/u/djinst1/sqllib/bin/db2acntp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""

local_tp:
    prof_name                                   = "zzserverint"
    tp_name                                     = "DB2INTERRUPT"
    tp_name_in_hex                              = no
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe          = "/u/djinst1/sqllib/bin/db2alttp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once

    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""
```

SNA/6000 Profile - part 5

```
local_tp:
    prof_name                                   = "djos2tp"
    tp_name                                     = "07F6C4C2"
    tp_name_in_hex                              = yes
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe              = "/u/djinst1/sqllib/bin/db2acntp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""

local_tp:
    prof_name                                   = "djos2int"
    tp_name                                     = "07F6E2D5"
    tp_name_in_hex                              = yes
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe              = "/u/djinst1/sqllib/bin/db2cnsm"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""
```

SNA/6000 Profile - part 6

```
link_station_token_ring:
    prof_name                                = "trlink"
    use_control_pt_xid                       = yes
    xid_node_id                              = "*"
    sna_dlc_profile_name                     = "tok0.00001"
    stop_on_inactivity                       = no
    time_out_value                           = 0
    LU_registration_supported                = no
    LU_registration_profile_name             = ""
    link_tracing                             = no
    trace_format                             = long
    access_routing_type                      = link_address
    remote_link_name                         = ""
    remote_link_address                      = 0x400008210200
    remote_sap                               = 0x04
    verify_adjacent_node                     = no
    net_id_of_adjacent_node                  = "USIBMSC"
    cp_name_of_adjacent_node                 = ""
    xid_node_id_of_adjacent_node             = "*"
    node_type_of_adjacent_node               = learn
    solicit_sscp_sessions                    = yes
    call_out_on_activation                   = yes
    activate_link_during_system_init         = yes
    activate_link_on_demand                  = no
    cp_cp_sessions_supported                 = yes
    cp_cp_session_support_required           = no
    adjacent_node_is_preferred_server        = no
    initial_tg_number                        = 0
    restart_on_normal_deactivation           = no
    restart_on_abnormal_deactivation         = no
    restart_on_activation                    = no
    TG_effective_capacity                    = 4300800
    TG_connect_cost_per_time                 = 0
    TG_cost_per_byte                         = 0
    TG_security                              = nonsecure
    TG_propagation_delay                     = lan
    TG_user_defined_1                        = 128
    TG_user_defined_2                        = 128
    TG_user_defined_3                        = 128
    comments                                 = ""

link_station_token_ring:
    prof_name                                = "trlnkyel"
    use_control_pt_xid                       = yes
    xid_node_id                              = "*"
    sna_dlc_profile_name                     = "tok0.00001"
    stop_on_inactivity                       = no
    time_out_value                           = 0
    LU_registration_supported                = no
    LU_registration_profile_name             = ""
    link_tracing                             = no
    trace_format                             = long
    access_routing_type                      = link_address
    remote_link_name                         = ""
    remote_link_address                      = 0x400052047184
    remote_sap                               = 0x04
    verify_adjacent_node                     = no
```

SNA/6000 Profile - part 7

```
    net_id_of_adjacent_node                 = "USIBMSC"
    cp_name_of_adjacent_node                = "SCA2109"
    xid_node_id_of_adjacent_node            = "*"
    node_type_of_adjacent_node              = learn
    solicit_sscp_sessions                   = yes
    call_out_on_activation                  = yes
    activate_link_during_system_init        = no
    activate_link_on_demand                 = no
    cp_cp_sessions_supported                = yes
    cp_cp_session_support_required          = no
    adjacent_node_is_preferred_server       = no
    initial_tg_number                       = 0
    restart_on_normal_deactivation          = no
    restart_on_abnormal_deactivation        = no
    restart_on_activation                   = no
    TG_effective_capacity                   = 4300800
    TG_connect_cost_per_time                = 0
    TG_cost_per_byte                        = 0
    TG_security                             = nonsecure
    TG_propagation_delay                    = lan
    TG_user_defined_1                       = 128
    TG_user_defined_2                       = 128
    TG_user_defined_3                       = 128
    comments                                = ""
sna_dlc_token_ring:
    prof_name                               = "tok0.00001"
    datalink_device_name                    = "tok0"
    force_timeout                           = 120
    user_defined_max_i_field                = no
    max_i_field_length                      = 30729
    max_active_link_stations                = 100
    num_reserved_inbound_activation         = 10
    num_reserved_outbound_activation        = 10
    transmit_window_count                   = 127
    dynamic_window_increment                = 1
    retransmit_count                        = 8
    receive_window_count                    = 1
    priority                                = 0
    inact_timeout                           = 48
    response_timeout                        = 4
    acknowledgement_timeout                 = 1
    link_name                               = "SEVLINK"
    local_sap                               = 0x04
    retry_interval                          = 60
    retry_limit                             = 20
```

SNA/6000 Profile - part 8

```
    dynamic_link_station_supported                = yes
    trace_base_listen_link_station                = no
    trace_base_listen_link_station_format         = long
    dynamic_lnk_solicit_sscp_sessions             = yes
    dynamic_lnk_cp_cp_sessions_supported          = yes
    dynamic_lnk_cp_cp_session_support_required    = no
    dynamic_lnk_TG_effective_capacity             = 4300800
    dynamic_lnk_TG_connect_cost_per_time          = 0
    dynamic_lnk_TG_cost_per_byte                  = 0
    dynamic_lnk_TG_security                       = nonsecure
    dynamic_lnk_TG_propagation_delay              = lan
    dynamic_lnk_TG_user_defined_1                 = 128
    dynamic_lnk_TG_user_defined_2                 = 128
    dynamic_lnk_TG_user_defined_3                 = 128
    comments                                      = ""
mode:
    prof_name                                     = "IBMRDBM"
    mode_name                                     = "IBMRDBM"
    max_sessions                                  = 200
    min_conwinner_sessions                        = 5
    min_conloser_sessions                         = 5
    auto_activate_limit                           = 0
    max_adaptive_receive_pacing_window            = 16
    receive_pacing_window                         = 3
    max_ru_size                                   = 2816
    min_ru_size                                   = 1024
    class_of_service_name                         = "#CONNECT"
    comments                                      = ""

mode:
    prof_name                                     = "IBMRDB"
    mode_name                                     = "IBMRDB"
    max_sessions                                  = 10
    min_conwinner_sessions                        = 5
    min_conloser_sessions                         = 5
    auto_activate_limit                           = 0
    max_adaptive_receive_pacing_window            = 16
    receive_pacing_window                         = 3
    max_ru_size                                   = 2816
    min_ru_size                                   = 1024
    class_of_service_name                         = "#CONNECT"
    comments                                      = ""

mode:
    prof_name                                     = "IBMROS2"
    mode_name                                     = "IBMROS2"
    max_sessions                                  = 10
    min_conwinner_sessions                        = 5
    min_conloser_sessions                         = 5
    auto_activate_limit                           = 0
    max_adaptive_receive_pacing_window            = 16
    receive_pacing_window                         = 3
    max_ru_size                                   = 2816
    min_ru_size                                   = 1024
    class_of_service_name                         = "#CONNECT"
    comments                                      = ""
```

Appendix F. CAE/2 Configuration File for APPC Connection

This appendix contains definitions made in an OS/2 client in Germany, where CAE/2 and CM/2 are installed to access DataJoiner using APPC protocol.

CM/2 NDF Profile - part 1

```
DEFINE_LOCAL_CP       FQ_CP_NAME(DEIBMIPF.APPCOS2 )
                      DESCRIPTION(Control Point in LAN)
                      CP_ALIAS(APPCOS2 )
                      NAU_ADDRESS(INDEPENDENT_LU)
                      NODE_TYPE(EN)
                      NODE_ID(X'05DE000C')
                      NW_FP_SUPPORT(NONE)
                      HOST_FP_SUPPORT(YES)
                      MAX_COMP_LEVEL(NONE)
                      MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK   LINK_NAME(LINK0001)
                      DESCRIPTION(3270 sessions)
                      ADJACENT_NODE_TYPE(NN)
                      PREFERRED_NN_SERVER(YES)
                      DLC_NAME(IBMTRNET)
                      ADAPTER_NUMBER(0)
                      DESTINATION_ADDRESS(X'40002020100104')
                      ETHERNET_FORMAT(NO)
                      CP_CP_SESSION_SUPPORT(YES)
                      SOLICIT_SSCP_SESSION(YES)
                      ACTIVATE_AT_STARTUP(YES)
                      USE_PUNAME_AS_CPNAME(NO)
                      LIMITED_RESOURCE(NO)
                      LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                      MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                      EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                      COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                      COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                      SECURITY(USE_ADAPTER_DEFINITION)
                      PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOGICAL_LINK   LINK_NAME(DATAJOIN)
                      DESCRIPTION(DJ via APPC)
                      FQ_ADJACENT_CP_NAME(DEIBMIPF.IPFBOEDJ)
                      ADJACENT_NODE_TYPE(LEARN)
                      DLC_NAME(IBMTRNET)
                      ADAPTER_NUMBER(0)
                      DESTINATION_ADDRESS(X'40001010101B04')
                      ETHERNET_FORMAT(NO)
                      CP_CP_SESSION_SUPPORT(NO)
                      SOLICIT_SSCP_SESSION(NO)
                      ACTIVATE_AT_STARTUP(YES)
                      USE_PUNAME_AS_CPNAME(NO)
                      LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                      LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
```

© Copyright IBM Corp. 1995

CM/2 NDF Profile - part 2

```
                              MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                              EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                              COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                              COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                              SECURITY(USE_ADAPTER_DEFINITION)
                              PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                              USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                              USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                              USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOCAL_LU               LU_NAME(DBSRV2  )
                              DESCRIPTION(Name of this client)
                              LU_ALIAS(DBSRV2  )
                              NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU             FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                              PARTNER_LU_ALIAS(IPFT1TRA)
                              PARTNER_LU_UNINTERPRETED_NAME(IPFBOEDJ)
                              MAX_MC_LL_SEND_SIZE(32767)
                              CONV_SECURITY_VERIFICATION(NO)
                              PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION    FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                              WILDCARD_ENTRY(NO)
                              FQ_OWNING_CP_NAME(DEIBMIPF.IPFBOEDJ)
                              LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE                   MODE_NAME(IBMRDBM )
                              COS_NAME(#CONNECT)
                              DEFAULT_RU_SIZE(NO)
                              MAX_RU_SIZE_UPPER_BOUND(1920)
                              RECEIVE_PACING_WINDOW(5)
                              MAX_NEGOTIABLE_SESSION_LIMIT(32767)
                              PLU_MODE_SESSION_LIMIT(10)
                              MIN_CONWINNERS_SOURCE(5)
                              COMPRESSION_NEED(PROHIBITED)
                              PLU_SLU_COMPRESSION(NONE)
                              SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS               IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                              DEFAULT_MODE_NAME(BLANK)
                              DEFAULT_LOCAL_LU_ALIAS(DBSRV2  )
                              MAX_MC_LL_SEND_SIZE(32767)
                              DIRECTORY_FOR_INBOUND_ATTACHES(*)
                              DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                              DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                              DEFAULT_TP_CONV_SECURITY_RQD(NO)
                              MAX_HELD_ALERTS(10);

DEFINE_CPIC_SIDE_INFO         SYMBOLIC_DESTINATION_NAME(SDBOEDJ )
                              PARTNER_LU_ALIAS(IPFT1TRA         )
                              MODE_NAME(IBMRDBM )
                              TP_NAME(OS2TP);

START_ATTACH_MANAGER;
```

Appendix G. DB2/2 v.1.x Configuration File for APPC Connection

This appendix contains definitions made in an OS/2 client in Germany, where DB2/2 v.1.x and CM/2 was installed to access DataJoiner using APPC protocol. The Communication Manager configuration file for this connection is listed below.

CM/2 NDF Profile - part 1

```
DEFINE_LOCAL_CP       FQ_CP_NAME(DEIBMIPF.IPFCP20B)
                      DESCRIPTION(Control Point in LAN)
                      CP_ALIAS(IPFCP20B)
                      NAU_ADDRESS(INDEPENDENT_LU)
                      NODE_TYPE(EN)
                      NODE_ID(X'05DE000B')
                      NW_FP_SUPPORT(NONE)
                      HOST_FP_SUPPORT(YES)
                      MAX_COMP_LEVEL(NONE)
                      MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK   LINK_NAME(DATAJOIN)
                      DESCRIPTION(DJ CONNECTION VIA APPC)
                      FQ_ADJACENT_CP_NAME(DEIBMIPF.IPFP221B)
                      ADJACENT_NODE_TYPE(LEARN)
                      DLC_NAME(IBMTRNET)
                      ADAPTER_NUMBER(0)
                      DESTINATION_ADDRESS(X'40001010101B04')
                      ETHERNET_FORMAT(NO)
                      CP_CP_SESSION_SUPPORT(NO)
                      ACTIVATE_AT_STARTUP(YES)
                      LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                      LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                      SOLICIT_SSCP_SESSION(NO)
                      MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                      USE_PUNAME_AS_CPNAME(NO)
                      EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                      COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                      COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                      SECURITY(USE_ADAPTER_DEFINITION)
                      PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_PARTNER_LU     FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                      DESCRIPTION(DATAJOINER IN BOEBLINGEN)
                      PARTNER_LU_ALIAS(IPFBOEDJ)
                      PARTNER_LU_UNINTERPRETED_NAME(IPFBOEDJ)
                      MAX_MC_LL_SEND_SIZE(32767)
                      CONV_SECURITY_VERIFICATION(NO)
                      PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                      DESCRIPTION(DATAJOINER IN BOEBLINGEN)
                      WILDCARD_ENTRY(NO)
                      FQ_OWNING_CP_NAME(DEIBMIPF.IPFP221B)
                      LOCAL_NODE_NN_SERVER(NO);
```

© Copyright IBM Corp. 1995

CM/2 NDF Profile - part 2

```
DEFINE_MODE      MODE_NAME(IBMRDBM )
                 DESCRIPTION(CONNECT)
                 COS_NAME(#CONNECT)
                 DEFAULT_RU_SIZE(YES)
                 RECEIVE_PACING_WINDOW(2)
                 MAX_NEGOTIABLE_SESSION_LIMIT(32767)
                 PLU_MODE_SESSION_LIMIT(20)
                 MIN_CONWINNERS_SOURCE(10)
                 COMPRESSION_NEED(PROHIBITED)
                 PLU_SLU_COMPRESSION(NONE)
                 SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(BLANK)
                 MAX_MC_LL_SEND_SIZE(32767)
                 DIRECTORY_FOR_INBOUND_ATTACHES(*)
                 DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                 DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                 DEFAULT_TP_CONV_SECURITY_RQD(NO)
                 MAX_HELD_ALERTS(10);

START_ATTACH_MANAGER;
```

Appendix H. DB2/2 v.2.1 Configuration File for APPC Connection

This appendix contains definitions made in an OS/2 client in Germany, where DB2/2 v.2.1 and CM/2 were installed to access DataJoiner using APPC protocol. The Communication Manager configuration file for this connection is listed below.

CM/2 NDF Profile - part 1

```
DEFINE_LOCAL_CP       FQ_CP_NAME(DEIBMIPF.IPFCP2OE)
                      DESCRIPTION(Control Point in LAN)
                      CP_ALIAS(IPFCP2OE)
                      NAU_ADDRESS(INDEPENDENT_LU)
                      NODE_TYPE(EN)
                      NODE_ID(X'05DE000E')
                      NW_FP_SUPPORT(NONE)
                      HOST_FP_SUPPORT(YES)
                      MAX_COMP_LEVEL(NONE)
                      MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK   LINK_NAME(DATAJOIN)
                      DESCRIPTION(RS/6000 via Token Ring)
                      FQ_ADJACENT_CP_NAME(DEIBMIPF.IPFP221B)
                      ADJACENT_NODE_TYPE(LEARN)
                      DLC_NAME(IBMTRNET)
                      ADAPTER_NUMBER(0)
                      DESTINATION_ADDRESS(X'40001010101B04')
                      ETHERNET_FORMAT(NO)
                      CP_CP_SESSION_SUPPORT(NO)
                      ACTIVATE_AT_STARTUP(YES)
                      LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                      LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                      SOLICIT_SSCP_SESSION(NO)
                      MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                      USE_PUNAME_AS_CPNAME(NO)
                      EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                      COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                      COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                      SECURITY(USE_ADAPTER_DEFINITION)
                      PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                      USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOCAL_LU       LU_NAME(IPFCLOE0)
                      DESCRIPTION(LOCAL LU6.2 )
                      LU_ALIAS(DBSRV4C )
                      NAU_ADDRESS(INDEPENDENT_LU);

DEFINE_PARTNER_LU     FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                      DESCRIPTION(DataJoiner on RS/6000)
                      PARTNER_LU_ALIAS(IPFBOEDJ)
                      PARTNER_LU_UNINTERPRETED_NAME(IPFBOEDJ)
                      MAX_MC_LL_SEND_SIZE(32767)
                      CONV_SECURITY_VERIFICATION(NO)
                      PARALLEL_SESSION_SUPPORT(YES);
```

© Copyright IBM Corp. 1995

CM/2 NDF Profile - part 2

```
DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(DEIBMIPF.IPFBOEDJ)
                            DESCRIPTION(DataJoiner on RS/6000)
                            WILDCARD_ENTRY(NO)
                            FQ_OWNING_CP_NAME(DEIBMIPF.IPFP221B)
                            LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE                 MODE_NAME(IBMRDBM )
                            DESCRIPTION(CONNECT)
                            COS_NAME(#CONNECT)
                            DEFAULT_RU_SIZE(YES)
                            RECEIVE_PACING_WINDOW(2)
                            MAX_NEGOTIABLE_SESSION_LIMIT(32767)
                            PLU_MODE_SESSION_LIMIT(20)
                            MIN_CONWINNERS_SOURCE(20)
                            COMPRESSION_NEED(PROHIBITED)
                            PLU_SLU_COMPRESSION(NONE)
                            SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS             IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                            DEFAULT_MODE_NAME(BLANK)
                            DEFAULT_LOCAL_LU_ALIAS(DBSRV4C )
                            MAX_MC_LL_SEND_SIZE(32767)
                            DIRECTORY_FOR_INBOUND_ATTACHES(*)
                            DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                            DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                            DEFAULT_TP_CONV_SECURITY_RQD(NO)
                            MAX_HELD_ALERTS(10);

DEFINE_TP                   SNA_SERVICE_TP_NAME(X'07',6DB)
                            DESCRIPTION(For DB2/2 V1.1 Connections)
                            PIP_ALLOWED(NO)
                            FILESPEC(notused)
                            CONVERSATION_TYPE(ANY_TYPE)
                            CONV_SECURITY_RQD(YES)
                            SYNC_LEVEL(EITHER)
                            TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
                            PROGRAM_TYPE(BACKGROUND)
                            INCOMING_ALLOCATE_QUEUE_DEPTH(255)
                            INCOMING_ALLOCATE_TIMEOUT(INFINITE)
                            RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP                   SNA_SERVICE_TP_NAME(X'07',6SN)
                            DESCRIPTION(For DB2/2 V1.1 Interrupts)
                            PIP_ALLOWED(NO)
                            FILESPEC(notused)
                            CONVERSATION_TYPE(ANY_TYPE)
                            CONV_SECURITY_RQD(YES)
                            SYNC_LEVEL(EITHER)
                            TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
                            PROGRAM_TYPE(BACKGROUND)
                            INCOMING_ALLOCATE_QUEUE_DEPTH(255)
                            INCOMING_ALLOCATE_TIMEOUT(INFINITE)
                            RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

CM/2 NDF Profile - part 3

```
DEFINE_TP               TP_NAME(DB2DUMMY)
                        DESCRIPTION(For V2.1)
                        PIP_ALLOWED(NO)
                        FILESPEC(notused)
                        CONVERSATION_TYPE(ANY_TYPE)
                        CONV_SECURITY_RQD(NO)
                        SYNC_LEVEL(EITHER)
                        TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
                        PROGRAM_TYPE(BACKGROUND)
                        INCOMING_ALLOCATE_QUEUE_DEPTH(255)
                        INCOMING_ALLOCATE_TIMEOUT(INFINITE)
                        RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP               TP_NAME(DB2INTERRUPT)
                        DESCRIPTION(For V2.1)
                        PIP_ALLOWED(NO)
                        FILESPEC(notused)
                        CONVERSATION_TYPE(ANY_TYPE)
                        CONV_SECURITY_RQD(NO)
                        SYNC_LEVEL(EITHER)
                        TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
                        PROGRAM_TYPE(BACKGROUND)
                        INCOMING_ALLOCATE_QUEUE_DEPTH(255)
                        INCOMING_ALLOCATE_TIMEOUT(INFINITE)
                        RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_CPIC_SIDE_INFO   SYMBOLIC_DESTINATION_NAME(SDBOEDJ )
                        PARTNER_LU_ALIAS(IPFBOEDJ         )
                        MODE_NAME(IBMRDBM )
                        TP_NAME(OS2TP);

START_ATTACH_MANAGER;
```

Appendix I. CAE/6000 Configuration File for APPC Connection

This appendix contains definitions made in an AIX client in Germany, where CAE/6000 v.1.x and SNA Server/6000 was installed to access DataJoiner using APPC protocol. The SNA Server/6000 configuration file for this connection is listed below (only the changed parts in our configuration are included in the list).

SNA Server/6000 Profiles - part 1

```
control_pt:
    prof_name                                   = "node_cp"
    xid_node_id                                 = "*"
    network_name                                = "DEIBMIPF"
    control_pt_name_alias                       = "IPFP221C"
    control_pt_name                             = "IPFP221C"
    control_pt_node_type                        = appn_end_node
    max_cached_trees                            = 500
    max_nodes_in_topology_database              = 500
    route_addition_resistance                   = 128
    comments                                    = ""

local_lu_lu6.2:
    prof_name                                   = "ddcslu01"
    local_lu_name                               = "IPFCL01C"
    local_lu_alias                              = "IPFCL01C"
    local_lu_dependent                          = no
    local_lu_address                            =
    sscp_id                                     = *
    link_station_prof_name                      = ""
    conversation_security_list_profile_name     = ""
    comments                                    = ""

partner_lu6.2_location:
    prof_name                                   = "ddcsitso1"
    fq_partner_lu_name                          = "DEIBMIPF.IPFBOEDJ"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "IPFCL01C"
    link_station_profile_name                   = "ddcsdjbb"
    comments                                    = ""

side_info:
    prof_name                                   = "itso1"
    local_lu_or_control_pt_alias                = "IPFCL01C"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "DEIBMIPF.IPFBOEDJ"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "OS2TP"
    comments                                    = ""
```

© Copyright IBM Corp. 1995

SNA Server/6000 Profiles - part 2

```
link_station_token_ring:
    prof_name                                = "ddcsdjbb"
    use_control_pt_xid                       = no
    xid_node_id                              = 0x071e001c
    sna_dlc_profile_name                     = "tok0.00001"
    stop_on_inactivity                       = no
    time_out_value                           = 0
    LU_registration_supported                = no
    LU_registration_profile_name             = ""
    link_tracing                             = no
    trace_format                             = long
    access_routing_type                      = link_address
    remote_link_name                         = ""
    remote_link_address                      = 0x40001010101b
    remote_sap                               = 0x04
    verify_adjacent_node                     = no
    net_id_of_adjacent_node                  = ""
    cp_name_of_adjacent_node                 = ""
    xid_node_id_of_adjacent_node             = "*"
    node_type_of_adjacent_node               = appn_end_node
    solicit_sscp_sessions                    = yes
    call_out_on_activation                   = yes
    activate_link_during_system_init         = yes
    activate_link_on_demand                  = no
    cp_cp_sessions_supported                 = yes
    cp_cp_session_support_required           = no
    adjacent_node_is_preferred_server        = no
    initial_tg_number                        = 0
    restart_on_normal_deactivation           = no
    restart_on_abnormal_deactivation         = no
    restart_on_activation                    = no
    TG_effective_capacity                    = 15974400
    TG_connect_cost_per_time                 = 0
    TG_cost_per_byte                         = 0
    TG_security                              = nonsecure
    TG_propagation_delay                     = lan
    TG_user_defined_1                        = 128
    TG_user_defined_2                        = 128
    TG_user_defined_3                        = 128
    comments                                 = ""

mode:
    prof_name                                = "IBMRDBM"
    mode_name                                = "IBMRDBM"
    max_sessions                             = 10
    min_conwinner_sessions                   = 5
    min_conloser_sessions                    = 5
    auto_activate_limit                      = 0
    max_adaptive_receive_pacing_window       = 16
    receive_pacing_window                    = 3
    max_ru_size                              = 2816
    min_ru_size                              = 1024
    class_of_service_name                    = "#CONNECT"
    comments                                 = ""
```

Appendix J. TCP/IP Correlations and Worksheets

This appendix contains examples of TCP/IP correlation between DataJoiner and clients, and DataJoiner and data sources. Worksheets are included for each example, to be used for planning your installation.

Figure 121 is an example showing DataJoiner and client correlation.

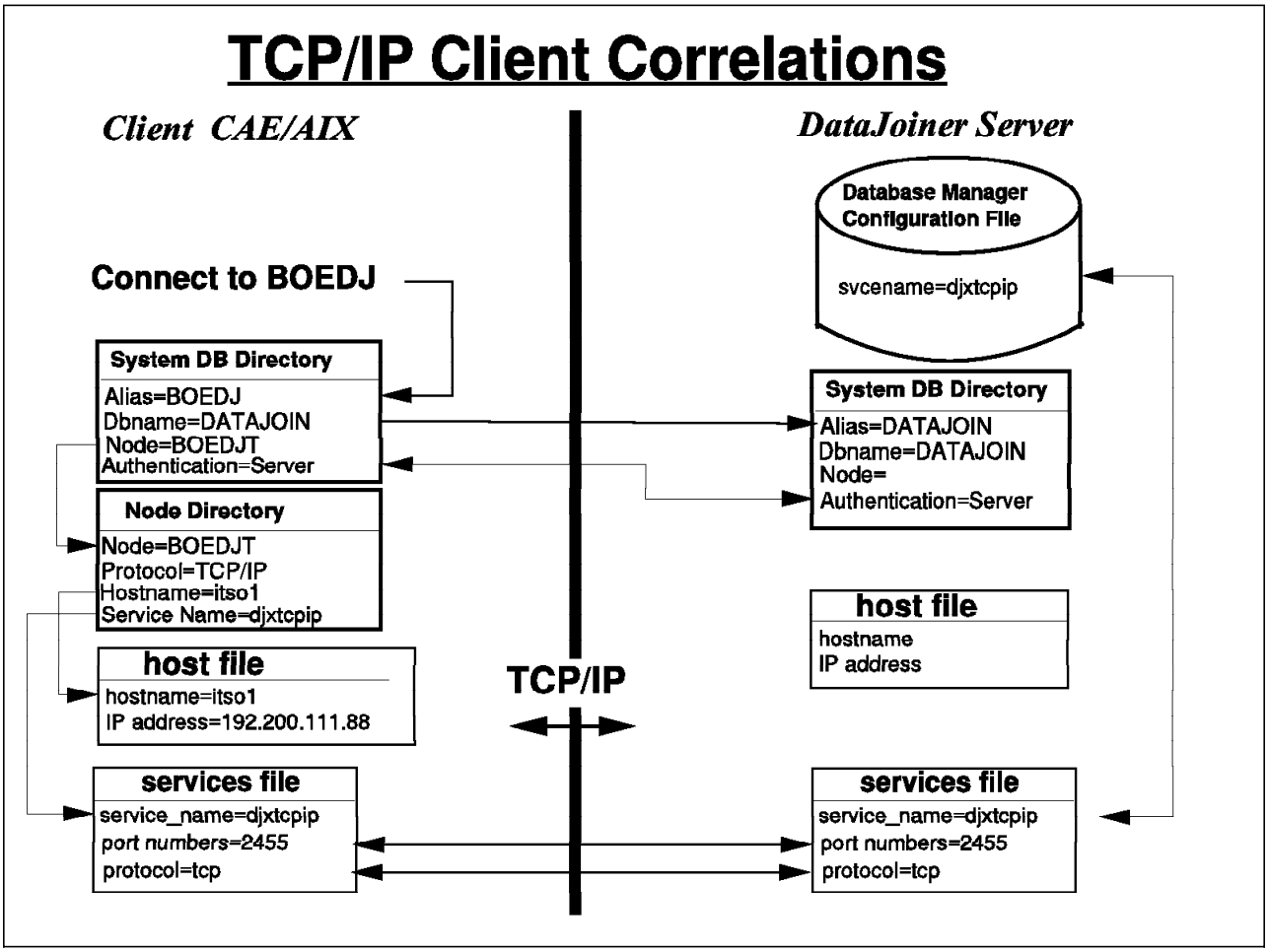


Figure 121. TCP/IP Client Correlation

Figure 122 on page 244 is a worksheet for DataJoiner and client correlation.

© Copyright IBM Corp. 1995

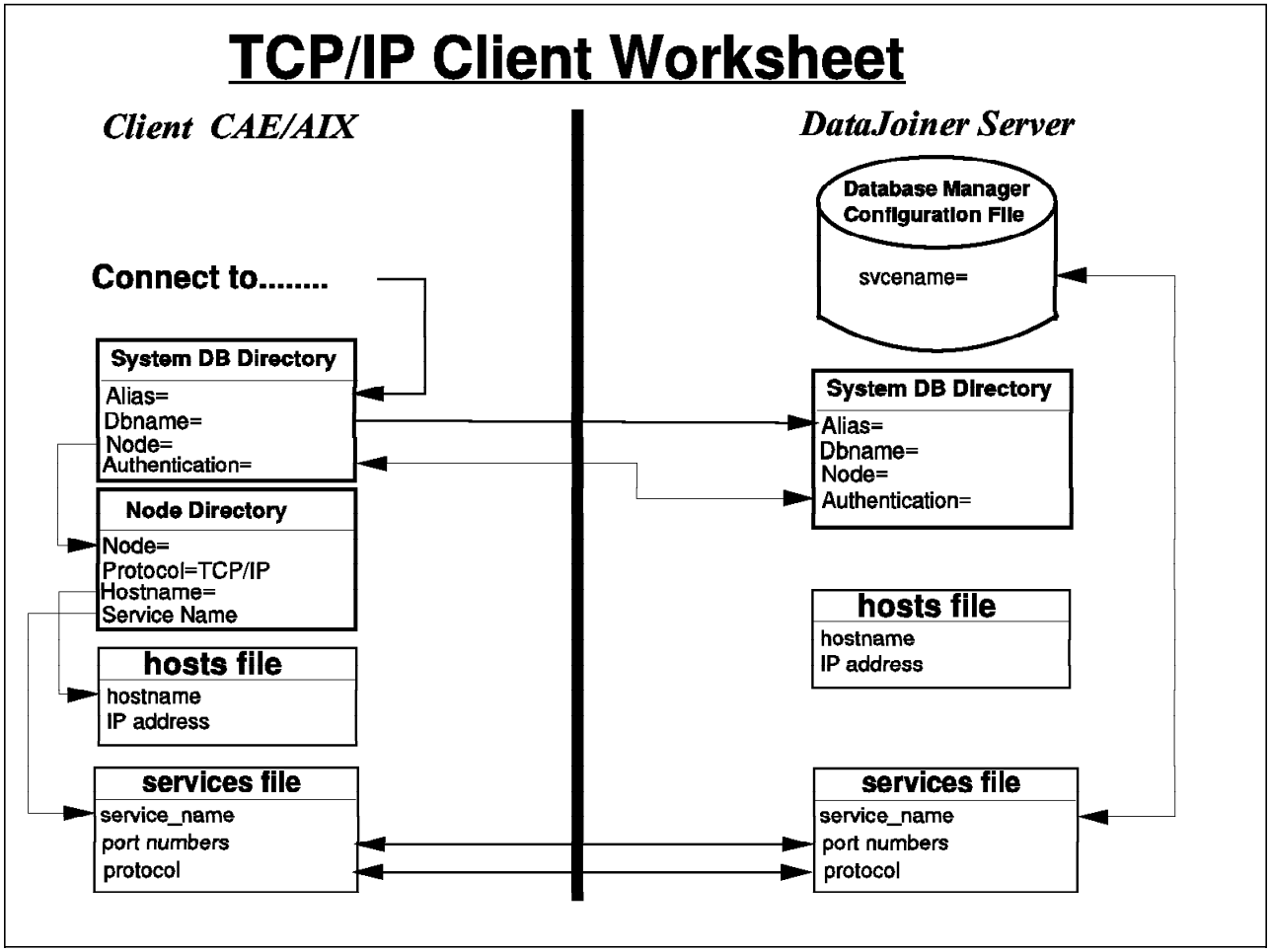


Figure 122. TCP/IP Client Correlation Worksheet

Figure 123 on page 245 is an example of the correlation between DataJoiner and data sources.

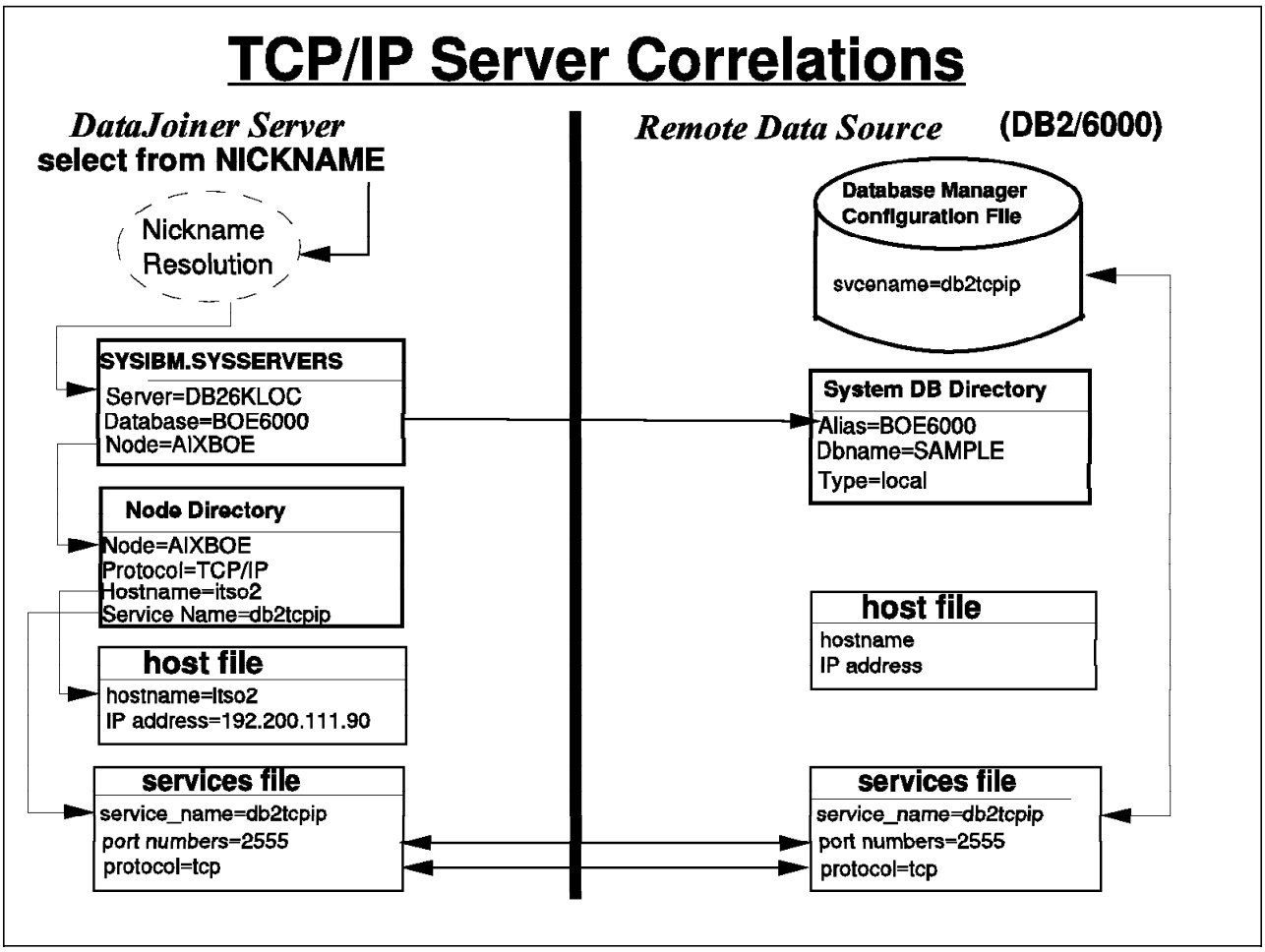


Figure 123. TCP/IP Server Correlation

Figure 124 on page 246 is a worksheet for DataJoiner and data sources correlation.

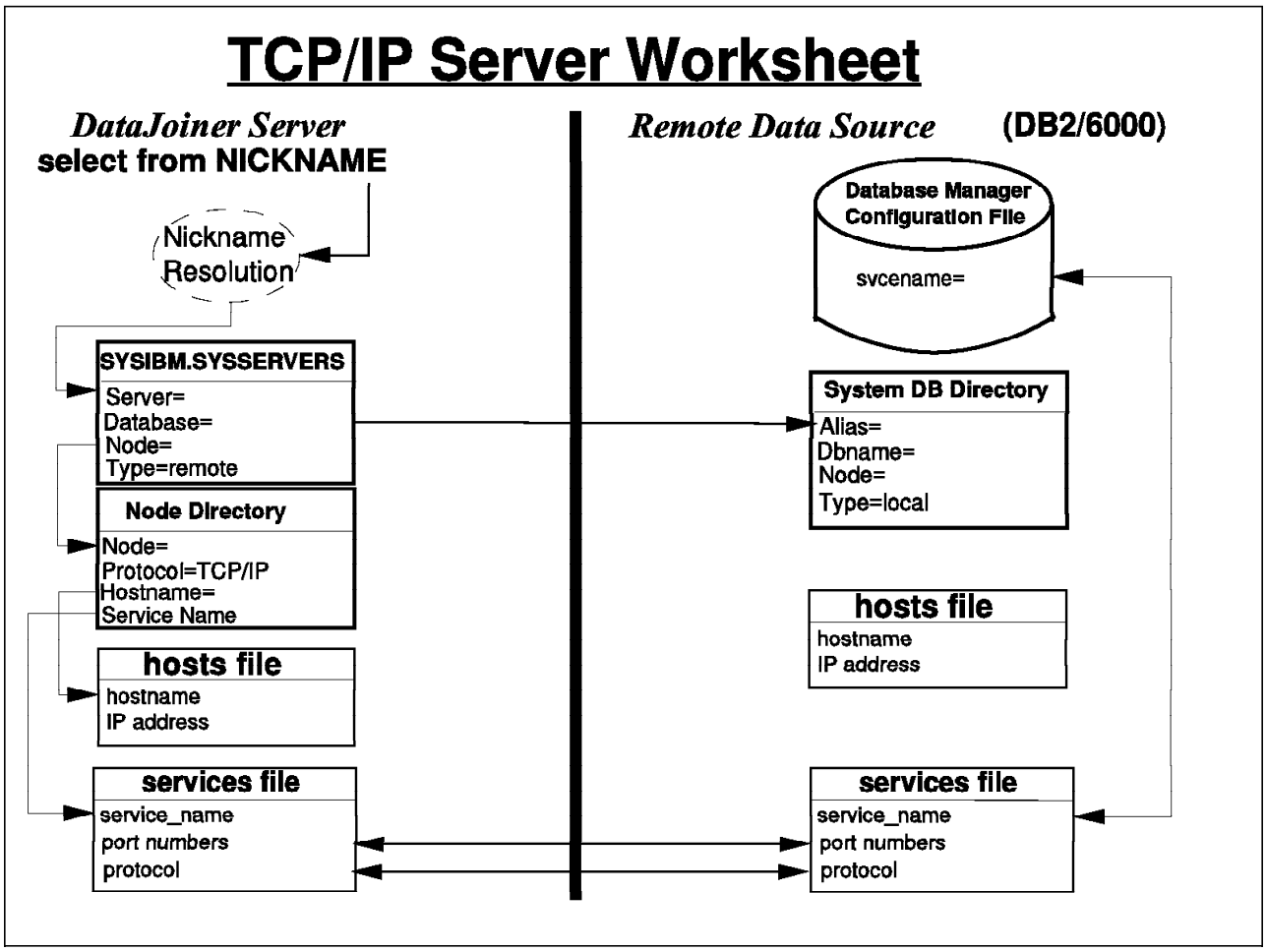


Figure 124. TCP/IP Server Correlation Worksheet

Appendix K. APPC (SNA) Correlations and Worksheets

This appendix contains examples of APPC Client correlations. A worksheet is included, for planning your installation.

Figure 125 shows the APPC client correlations.

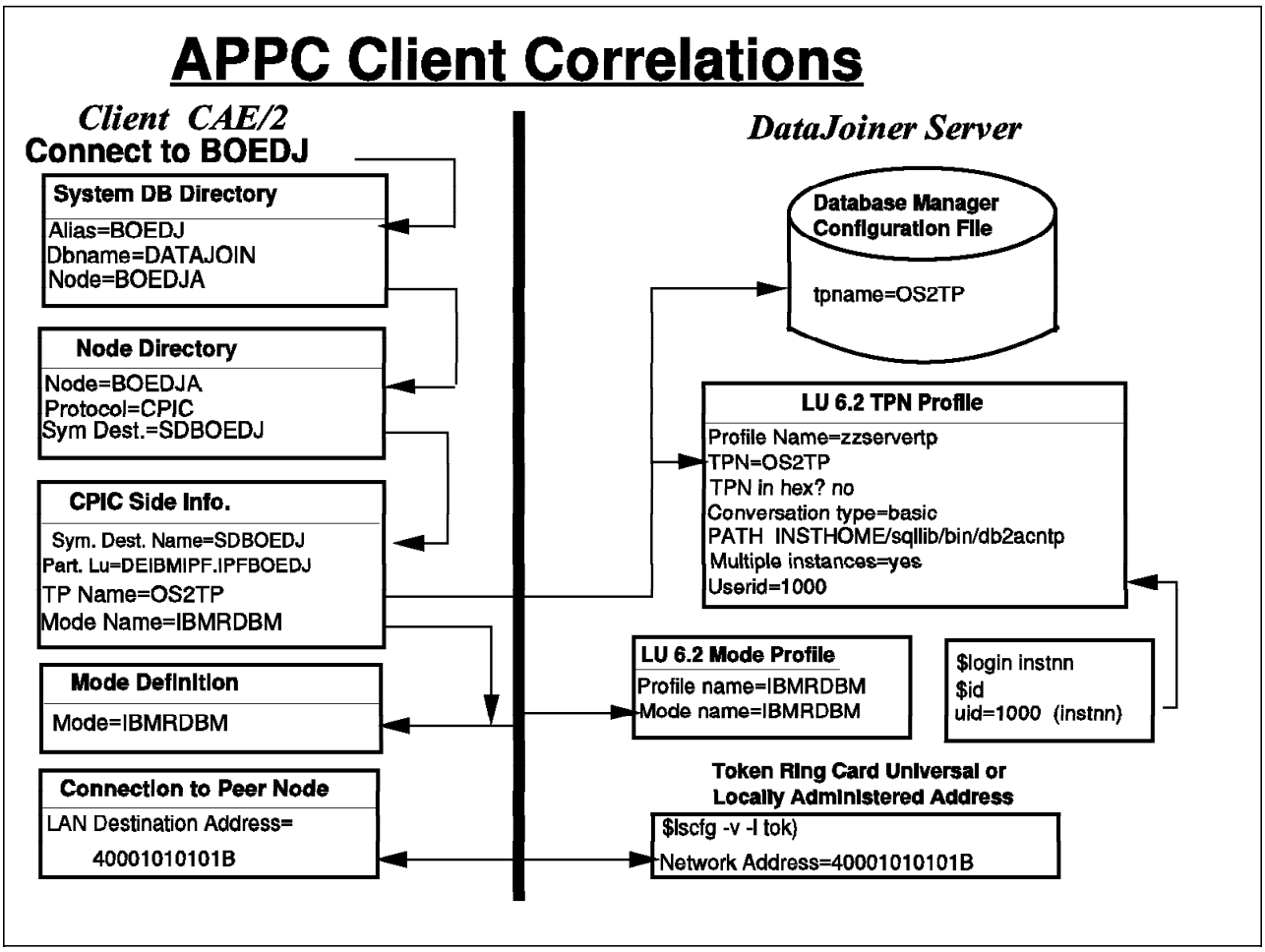


Figure 125. APPC Client Correlations

Figure 126 on page 248 is a worksheet for the APPC Client correlations.

© Copyright IBM Corp. 1995

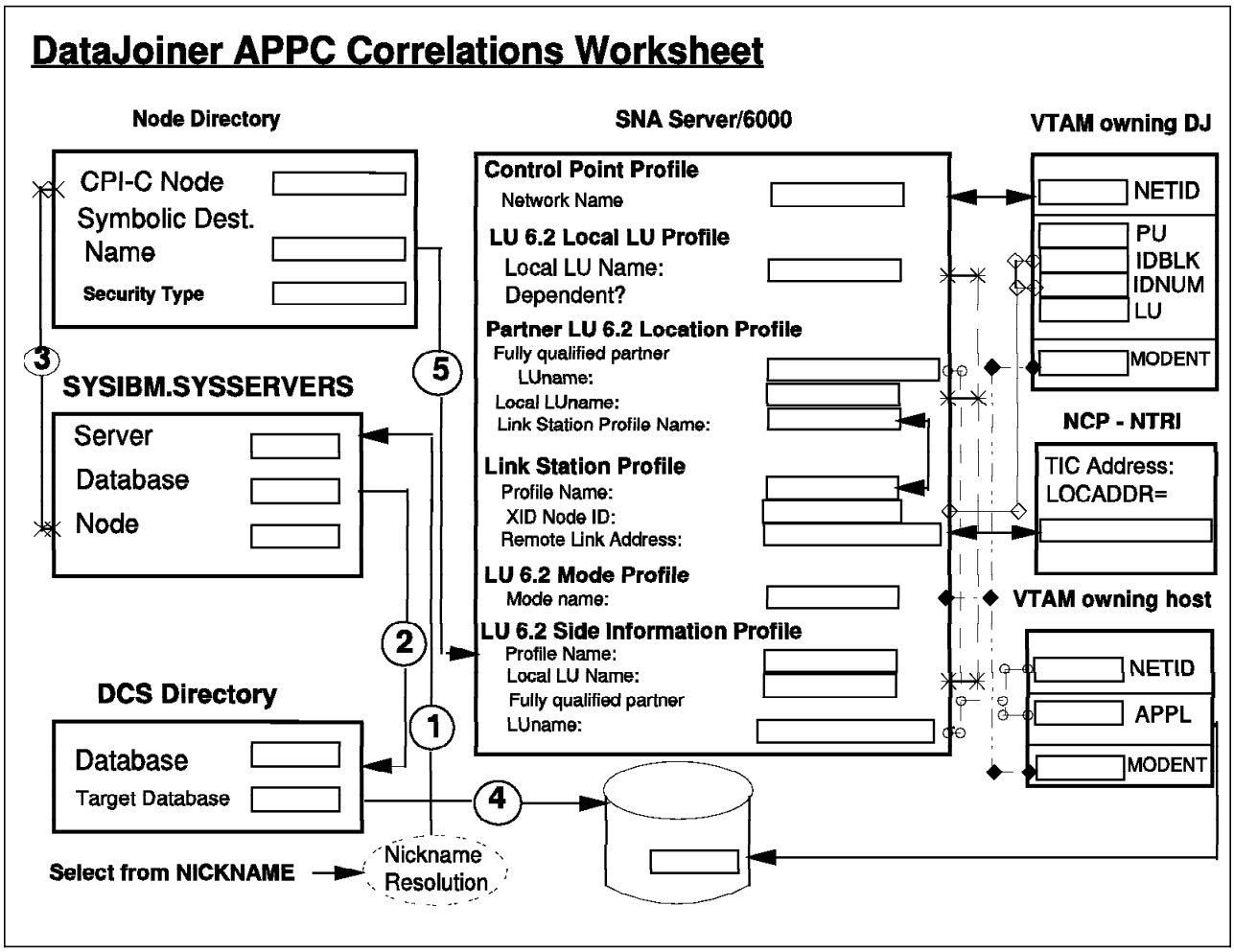


Figure 126. APPC Client Correlation Worksheet

Appendix L. AIX Definitions in the Severn System

This appendix contains definitions made in the Severn system in San Jose, where DataJoiner is installed.

SNA/6000 Profile - part 1

```
sna:
    prof_name                                       = "sna"
    max_sessions                                    = 200
    max_conversations                               = 200
    restart_action                                  = once
    rrm_enabled                                     = no
    dynamic_inbound_partner_lu_definitions_allowed = yes
    standard_output_device                          = "/dev/console"
    standard_error_device                           = "/var/sna/sna.stderr"
    nmvt_action_when_no_nmvt_process                = reject
    comments                                        = ""

control_pt:
    prof_name                                       = "node_cp"
    xid_node_id                                     = 0x071a2085
    network_name                                    = "USIBMSC"
    control_pt_name_alias                           = "SCA2085"
    control_pt_name                                 = "SCA2085"
    control_pt_node_type                            = appn_end_node
    max_cached_trees                                = 500
    max_nodes_in_topology_database                  = 500
    route_addition_resistance                       = 128
    comments                                        = ""

local_lu_lu6.2:
    prof_name                                       = "SCA2085I"
    local_lu_name                                   = "SCA2085I"
    local_lu_alias                                  = "SCA2085I"
    local_lu_dependent                              = no
    local_lu_address                                =
    sscp_id                                         = *
    link_station_prof_name                          = ""
    conversation_security_list_profile_name         = ""
    comments                                        = ""

partner_lu6.2_location:
    prof_name                                       = "DB3APLU"
    fq_partner_lu_name                              = "USIBMSC.SCLUDB3A"
    partner_location_method                         = link_station
    fq_partner_owning_cp_name                       = ""
    local_node_is_network_server_for_len_node       = no
    fq_node_server_name                             = ""
    local_lu_name                                   = "SCA2085I"
    link_station_profile_name                       = "trlink"
    comments                                        = ""
```

© Copyright IBM Corp. 1995

SNA/6000 Profile - part 2

```
partner_lu6.2_location:
    prof_name                                   = "SQLDSPLU"
    fq_partner_lu_name                          = "DEIBMIPF.IPFA2GL4"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "SQL/DS GERMANY"

partner_lu6.2_location:
    prof_name                                   = "DB23PLU"
    fq_partner_lu_name                          = "USIBMSC.LUDB23"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "DB2 V2.3 Pok"

partner_lu6.2_location:
    prof_name                                   = "DJYELPLU"
    fq_partner_lu_name                          = "USIBMSC.SCA2109I"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlnkyel"
    comments                                    = "DJ in yellow"

partner_lu6.2_location:
    prof_name                                   = "DB2VSPLU"
    fq_partner_lu_name                          = "DEIBMIPF.IPFA21CD"
    partner_location_method                     = link_station
    fq_partner_owning_cp_name                   = ""
    local_node_is_network_server_for_len_node   = no
    fq_node_server_name                         = ""
    local_lu_name                               = "SCA2O85I"
    link_station_profile_name                   = "trlink"
    comments                                    = "DB2/VSE GERMANY"

side_info:
    prof_name                                   = "db23atpn"
    local_lu_or_control_pt_alias                = "SCA2O85I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.SCLUDB3A"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = yes
    remote_tp_name                              = "07F6C4C2"
    comments                                    = ""
```

SNA/6000 Profile - part 3

```
side_info:
    prof_name                                   = "DB2VMTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "DEIBMIPF.IPFA2GL4"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "S34VMDBO"
    comments                                    = "SQL/DS GERMANY"

side_info:
    prof_name                                   = "db223tpn"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.LUDB23"
    mode_name                                   = "IBMRDB"
    remote_tp_name_in_hex                       = yes
    remote_tp_name                              = "07F6C4C2"
    comments                                    = "DB2 V2 Pok"

side_info:
    prof_name                                   = "DJYELTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "USIBMSC.SCA2109I"
    mode_name                                   = "IBMRDB"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "zzservertp"
    comments                                    = "DJ in yellow"

side_info:
    prof_name                                   = "DB2VSTPN"
    local_lu_or_control_pt_alias                = "SCA2085I"
    partner_lu_alias                            = ""
    fq_partner_lu_name                          = "DEIBMIPF.IPFA21CD"
    mode_name                                   = "IBMRDBM"
    remote_tp_name_in_hex                       = no
    remote_tp_name                              = "TPN1"
    comments                                    = "DB2/VSE GERMANY"
```

SNA/6000 Profile - part 4

```
local_tp:
    prof_name                                   = "zzservertp"
    tp_name                                     = "zzservertp"
    tp_name_in_hex                              = no
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe                 = "/u/djinst1/sqllib/bin/db2acntp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""

local_tp:
    prof_name                                   = "zzserverint"
    tp_name                                     = "DB2INTERRUPT"
    tp_name_in_hex                              = no
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe               = "/u/djinst1/sqllib/bin/db2alttp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""
```

SNA/6000 Profile - part 5

```
local_tp:
    prof_name                                   = "djos2tp"
    tp_name                                     = "07F6C4C2"
    tp_name_in_hex                              = yes
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe                 = "/u/djinst1/sqllib/bin/db2acntp"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""

local_tp:
    prof_name                                   = "djos2int"
    tp_name                                     = "07F6E2D5"
    tp_name_in_hex                              = yes
    pip_data_present                            = no
    pip_data_subfields_number                   = 0
    conversation_type                           = basic
    sync_level                                  = none/confirm
    resource_security_level                     = none
    resource_access_list_profile_name           = ""
    full_path_tp_exe                 = "/u/djinst1/sqllib/bin/db2cnsm"
    multiple_instances                          = yes
    user_id                                     = 202
    server_synonym_name                         = ""
    restart_action                              = once
    communication_type                          = signals
    ipc_queue_key                               = 0
    standard_input_device                       = "/dev/console"
    standard_output_device                      = "/dev/console"
    standard_error_device                       = "/dev/console"
    comments                                    = ""
```

SNA/6000 Profile - part 6

```
link_station_token_ring:
    prof_name                                = "trlink"
    use_control_pt_xid                       = yes
    xid_node_id                              = "*"
    sna_dlc_profile_name                     = "tok0.00001"
    stop_on_inactivity                       = no
    time_out_value                           = 0
    LU_registration_supported                = no
    LU_registration_profile_name             = ""
    link_tracing                             = no
    trace_format                             = long
    access_routing_type                      = link_address
    remote_link_name                         = ""
    remote_link_address                      = 0x400008210200
    remote_sap                               = 0x04
    verify_adjacent_node                     = no
    net_id_of_adjacent_node                  = "USIBMSC"
    cp_name_of_adjacent_node                 = ""
    xid_node_id_of_adjacent_node             = "*"
    node_type_of_adjacent_node               = learn
    solicit_sscp_sessions                    = yes
    call_out_on_activation                   = yes
    activate_link_during_system_init         = yes
    activate_link_on_demand                  = no
    cp_cp_sessions_supported                 = yes
    cp_cp_session_support_required           = no
    adjacent_node_is_preferred_server        = no
    initial_tg_number                        = 0
    restart_on_normal_deactivation           = no
    restart_on_abnormal_deactivation         = no
    restart_on_activation                    = no
    TG_effective_capacity                    = 4300800
    TG_connect_cost_per_time                 = 0
    TG_cost_per_byte                         = 0
    TG_security                              = nonsecure
    TG_propagation_delay                     = lan
    TG_user_defined_1                        = 128
    TG_user_defined_2                        = 128
    TG_user_defined_3                        = 128
    comments                                 = ""
```

SNA/6000 Profile - part 7

```
link_station_token_ring:
    prof_name                                = "trlnkyel"
    use_control_pt_xid                       = yes
    xid_node_id                              = "*"
    sna_dlc_profile_name                     = "tok0.00001"
    stop_on_inactivity                       = no
    time_out_value                           = 0
    LU_registration_supported                = no
    LU_registration_profile_name             = ""
    link_tracing                             = no
    trace_format                             = long
    access_routing_type                      = link_address
    remote_link_name                         = ""
    remote_link_address                      = 0x400052047184
    remote_sap                               = 0x04
    verify_adjacent_node                     = no
    net_id_of_adjacent_node                  = "USIBMSC"
    cp_name_of_adjacent_node                 = "SCA2109"
    xid_node_id_of_adjacent_node             = "*"
    node_type_of_adjacent_node               = learn
    solicit_sscp_sessions                    = yes
    call_out_on_activation                   = yes
    activate_link_during_system_init         = no
    activate_link_on_demand                  = no
    cp_cp_sessions_supported                 = yes
    cp_cp_session_support_required           = no
    adjacent_node_is_preferred_server        = no
    initial_tg_number                        = 0
    restart_on_normal_deactivation           = no
    restart_on_abnormal_deactivation         = no
    restart_on_activation                    = no
    TG_effective_capacity                    = 4300800
    TG_connect_cost_per_time                 = 0
    TG_cost_per_byte                         = 0
    TG_security                              = nonsecure
    TG_propagation_delay                     = lan
    TG_user_defined_1                        = 128
    TG_user_defined_2                        = 128
    TG_user_defined_3                        = 128
    comments                                 = ""
```

SNA/6000 Profile - part 8

```
sna_dlc_token_ring:
    prof_name                                   = "tok0.00001"
    datalink_device_name                        = "tok0"
    force_timeout                               = 120
    user_defined_max_i_field                    = no
    max_i_field_length                          = 30729
    max_active_link_stations                    = 100
    num_reserved_inbound_activation             = 10
    num_reserved_outbound_activation            = 10
    transmit_window_count                       = 127
    dynamic_window_increment                    = 1
    retransmit_count                            = 8
    receive_window_count                        = 1
    priority                                    = 0
    inact_timeout                               = 48
    response_timeout                            = 4
    acknowledgement_timeout                     = 1
    link_name                                   = "SEVLINK"
    local_sap                                   = 0x04
    retry_interval                              = 60
    retry_limit                                 = 20
    dynamic_link_station_supported              = yes
    trace_base_listen_link_station              = no
    trace_base_listen_link_station_format       = long
    dynamic_lnk_solicit_sscp_sessions           = yes
    dynamic_lnk_cp_cp_sessions_supported        = yes
    dynamic_lnk_cp_cp_session_support_required  = no
    dynamic_lnk_TG_effective_capacity           = 4300800
    dynamic_lnk_TG_connect_cost_per_time        = 0
    dynamic_lnk_TG_cost_per_byte                = 0
    dynamic_lnk_TG_security                     = nonsecure
    dynamic_lnk_TG_propagation_delay            = lan
    dynamic_lnk_TG_user_defined_1               = 128
    dynamic_lnk_TG_user_defined_2               = 128
    dynamic_lnk_TG_user_defined_3               = 128
    comments                                    = ""

mode:
    prof_name                                   = "IBMRDBM"
    mode_name                                   = "IBMRDBM"
    max_sessions                                = 200
    min_conwinner_sessions                      = 5
    min_conloser_sessions                       = 5
    auto_activate_limit                         = 0
    max_adaptive_receive_pacing_window          = 16
    receive_pacing_window                       = 3
    max_ru_size                                 = 2816
    min_ru_size                                 = 1024
    class_of_service_name                       = "#CONNECT"
    comments                                    = ""
```

SNA/6000 Profile - part 9

```
mode:
    prof_name                                = "IBMRDB"
    mode_name                                = "IBMRDB"
    max_sessions                             = 10
    min_conwinner_sessions                   = 5
    min_conloser_sessions                    = 5
    auto_activate_limit                      = 0
    max_adaptive_receive_pacing_window       = 16
    receive_pacing_window                    = 3
    max_ru_size                              = 2816
    min_ru_size                              = 1024
    class_of_service_name                    = "#CONNECT"
    comments                                 = ""

mode:
    prof_name                                = "IBMROS2"
    mode_name                                = "IBMROS2"
    max_sessions                             = 10
    min_conwinner_sessions                   = 5
    min_conloser_sessions                    = 5
    auto_activate_limit                      = 0
    max_adaptive_receive_pacing_window       = 16
    receive_pacing_window                    = 3
    max_ru_size                              = 2816
    min_ru_size                              = 1024
    class_of_service_name                    = "#CONNECT"
    comments                                 = ""
```

Database Manager Configuration

```
get database manager configuration

          Database Manager Configuration

 Database manager configuration release level  = 0x0500
 Node type                                      = Server with remote clients

 Service name                                (SVCENAME) = severndj
 Transaction program name                      (TPNAME) = zzservertp

 Max requester I/O block size (bytes)         (RQRIOBLK) = 4096
 Max server I/O block size (bytes)            (SVRIOBLK) = 4096
 Communication heap size (4KB)               (COMHEAPSZ) = 128
 Remote services heap size (4KB)              (RSHEAPSZ) = 128
 Sort heap threshold (4KB)                   (SHEAPTHRES) = 25000
 Application support layer heap size (4KB)    (ASLHEAPSZ) = 100

 Max no. of existing agents                   (MAXAGENTS) = 144
 Max no. of concurrent agents                (MAXCAGENTS) = MAXAGENTS
 Max no. of concurrently active databases         (NUMDB) = 2

 Application cleanup interval (ms)           (CUINTERVAL) = 5000
 Keep DARI process                             (KEEPDARI) = YES
 Max. no. of DARI processes                     (MAXDARI) = 1
 Priority of agents                            (AGENTPRI) = 59
 Database monitor SQL statement size (bytes) (SQLSTMTSZ) = 0
 Index re-creation time                         (INDEXREC) = RESTART
 Default database path                         (DFTDBPATH) = /home/djinst1

 Backup buffer default size (4KB)             (BACKBUFSZ) = 1024
 Restore buffer default size (4KB)            (RESTBUFSZ) = 1024
```

Node Directory - part 1

```
list node directory

 Node Directory

 Number of entries in the directory = 12

Node 1 entry:

 Node name                      = TODB223
 Comment                        =
 Protocol                       = CPIC
 Symbolic destination name      = db223tpn
 Security type                  = PROGRAM

Node 2 entry:

 Node name                      = TODB23A
 Comment                        =
 Protocol                       = CPIC
 Symbolic destination name      = db23atpn
 Security type                  = PROGRAM

Node 3 entry:

 Node name                      = TODB2VSE
 Comment                        =
 Protocol                       = CPIC
 Symbolic destination name      = DB2VSTPN
 Security type                  = PROGRAM

Node 4 entry:

 Node name                      = TODJ2SEV
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = severn
 Service name                   = sevdj2

Node 5 entry:

 Node name                      = TODJGER
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = itso1
 Service name                   = djxtcpip

Node 6 entry:

 Node name                      = TODJSEV
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = severn
 Service name                   = severndj
```

Node Directory - part 2

```
Node 7 entry:

 Node name                      = TODJYEL
 Comment                        =
 Protocol                       = CPIC
 Symbolic destination name      = DJYELTPN
 Security type                  = PROGRAM

Node 8 entry:

 Node name                      = TODJYELL
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = yellow
 Service name                   = djmn

Node 9 entry:

 Node name                      = TODJZEUS
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = zeus
 Service name                   = iwdj01

Node 10 entry:

 Node name                      = TOJAVA
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = java
 Service name                   = java

Node 11 entry:

 Node name                      = TOSQLDS
 Comment                        =
 Protocol                       = CPIC
 Symbolic destination name      = DB2VMTPN
 Security type                  = PROGRAM

Node 12 entry:

 Node name                      = TOV2YELL
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = yellow
 Service name                   = db2v2c
```

System Database Directory - part 1

```
list database directory

 System Database Directory

 Number of entries in the directory = 16

Database 1 entry:

 Database alias                  = DBDJ2SEV
 Database name                   = DBINST2
 Local database directory        =
 Database directory              =
 Node name                       = TODJ2SEV
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = SERVER

Database 2 entry:

 Database alias                  = DB23A
 Database name                   = DB23A
 Local database directory        =
 Database directory              =
 Node name                       = TODB23A
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = DCS

Database 3 entry:

 Database alias                  = YELLOWV2
 Database name                   = SAMPLE
 Local database directory        =
 Database directory              =
 Node name                       = TOV2YELL
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = CLIENT

Database 4 entry:

 Database alias                  = DB2VSE
 Database name                   = DB2VSE
 Local database directory        =
 Database directory              =
 Node name                       = TODB2VSE
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = DCS
```

System Database Directory - part 2

```
Database 5 entry:

 Database alias                 = DBDJGER
 Database name                  = DATAJOIN
 Local database directory       =
 Database directory             =
 Node name                      = TODJGER
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 6 entry:

 Database alias                 = DBJAVA
 Database name                  = SAMPLE
 Local database directory       =
 Database directory             =
 Node name                      = TOJAVA
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 7 entry:

 Database alias                 = YELLOWDJ
 Database name                  = DBINST1
 Local database directory       =
 Database directory             =
 Node name                      = TODJYELL
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Remote
 Authentication                 = DCS

Database 8 entry:

 Database alias                 = DBINST1
 Database name                  = DBINST1
 Local database directory       = /home/djinst1
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Indirect
 Authentication                 = SERVER
```

System Database Directory - part 3

```
Database 9 entry:

 Database alias                  = ZEUSDJ
 Database name                   = DEMO
 Local database directory        =
 Database directory              =
 Node name                       = TODJZEUS
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = DCS

Database 10 entry:

 Database alias                  = TEST
 Database name                   = TEST
 Local database directory        = /usr/usrt001
 Database directory              =
 Node name                       =
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Indirect
 Authentication                  = SERVER

Database 11 entry:

 Database alias                  = DBDJYEL
 Database name                   = DBINST1
 Local database directory        =
 Database directory              =
 Node name                       = TODJYEL
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Remote
 Authentication                  = SERVER

Database 12 entry:

 Database alias                  = TPCCLOC
 Database name                   = TPCCLOC
 Local database directory        = /usr/tpccloc
 Database directory              =
 Node name                       =
 Database release level          = 5.00
 Comment                         =
 Directory entry type            = Indirect
 Authentication                  = SERVER
```

System Database Directory - part 4

```
Database 13 entry:

 Database alias                 = DB223
 Database name                  = DB223
 Local database directory       =
 Database directory             =
 Node name                      = TODB223
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Remote
 Authentication                 = SERVER

Database 14 entry:

 Database alias                 = TPCC
 Database name                  = TPCC
 Local database directory       = /usr/tpcc
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Indirect
 Authentication                 = SERVER

Database 15 entry:

 Database alias                 = SAMPLE
 Database name                  = SAMPLE
 Local database directory       = /home/djinst1
 Database directory             =
 Node name                      =
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Indirect
 Authentication                 = CLIENT

Database 16 entry:

 Database alias                 = DBSQLDS
 Database name                  = DBSQLDS
 Local database directory       =
 Database directory             =
 Node name                      = TOSQLDS
 Database release level         = 5.00
 Comment                        =
 Directory entry type           = Remote
 Authentication                 = DCS
```

DCS Directory - part 1

```
list dcs directory

 Database Connection Services (DCS) Directory

 Number of entries in the directory = 4

DCS 1 entry:

 Local database name                = DB223
 Target database name               = DB2CENTDIST
 Application requestor name         =
 DCS parameters                     =
 Comment                            =
 DCS directory release level        = 0x0100

DCS 2 entry:

 Local database name                = DB23A
 Target database name               = CENTSJC
 Application requestor name         =
 DCS parameters                     =
 Comment                            =
 DCS directory release level        = 0x0100

DCS 3 entry:

 Local database name                = DB2VSE
 Target database name               = S34VSDB1
 Application requestor name         =
 DCS parameters                     =
 Comment                            =
 DCS directory release level        = 0x0100

DCS 4 entry:

 Local database name                = DBSQLDS
 Target database name               = S34VMDBO
 Application requestor name         =
 DCS parameters                     =
 Comment                            =
 DCS directory release level        = 0x0100
```

DCS Directory - part 1

SYSIBM.SYSSERVERS

SERVER	NODE	DBNAME	TYPE	VERSION	PROTOCOL	PASSWORD
SYBASE1	SybaseMVP	test_db	SYBASE	4.3	openclient	Y
DJ2SEV	TODJ2SEV	DBINST2	DB2/6000	1.0	jra	Y
DB223	TODB223	DB2CENTDIST	DB2/MVS	2.3	drda	Y
DB23A	TODB23A	CENTSJC	DB2/MVS	3.1	drda	Y
ORACLE1	mvpo		ORACLE	7.0	sqlnet	Y
JAVA	TOJAVA	sample	DB2/2	2.1	jra	Y

SYSIBM.SYSREMOTEUSERS

AUTHID	SERVER	REMOTE_AUTHID	REMOTE_PW
DJINST1	SYBASE1	iwdj01	x'383B29FD57DC7D13'
DJINST1	DJ2SEV	iwdj01	x'D6B8F87A6C0CF1D5'
DJINST1	DB223	iwdj01	x'D6B8F87A6C0CF1D5'
DJINST1	DB23A	iwdj01	x'D6B8F87A6C0CF1D5'
DJINST1	ORACLE1	iwdj01	x'383B29FD57DC7D13'
IWDJ02	DB23A	IWDJ02	x'F7D3A2B31F30DCA8'
IWDJ02	SYBASE1	iwdj02	x'EFF5F883773CF075'
DJINST1	JAVA	userid	x'60D85BFE82856FC8'
IWDJ02	ORACLE1	iwdj02	x'EFF5F883773CF075'
IWDJ02	DB223	IWDJ02	x'F7D3A2B31F30DCA8'

List of Abbreviations

AIX	Advanced Interactive Executive (operating system for IBM RISC System/6000)
APA	all points addressable
API	application programming interface
APPC	Advanced Program-to-Program Communication
APPN	Advanced Peer-to-Peer Networking
BSDS	bootstrap data set
CAE	Client Application Enabler
CD-ROM	compact disk read-only memory
CLI	call-level interface
CLP	command-line processor
CM	Communications Manager
CP	control point
CPI-C	common programming interface for communications
CPU	central processing unit
DBA	database administrator
DBMS	database management system
DCL	data control language
DCS	database connection services
DDCS/6000	Distributed Database Connection Services for AIX/6000
DDL	data definition language
DLC	data link control
DRDA	Distributed Relational Database Architecture
ESA	enterprise systems architecture
I/O	input/output
IMS	Information Management System
IP	Internet protocol
IPL	initial program load
ISDN	Integrated Services Digital Network
ITSO	International Technical Support Organization
JRA	journaled relational access
LAN	local area network
LU	logical unit (SNA)
MPN	Multiprotocol Network Program
MVS	multiple virtual storage (S/390 operating system)
ODBC	open database connectivity
PTF	program temporary fix
PU	physical unit (SNA)
RISC	reduced instruction set computer
SDK	software developer's kit
SMIT	System Management Interface Tool
SNA	systems network architecture
SQL	structured query language
SWVPD	Software Vital Product Data
sysadm	system administrator (level of authority in DB2)
TCP/IP	Transmission Control Protocol/Internet Protocol
TIC	token ring interface card
TP	transaction program
TPN	transaction program name
UPM	User Profile Management (component of OS/2)
URL	Uniform Resource Locator
VM	virtual memory
VQW	Visualizer Query for Windows
VSAM	virtual storage access method
VSE	virtual storage extended
VTAM	virtual telecommunications access method
WAN	wide area network

© Copyright IBM Corp. 1995

Index

A

C

© Copyright IBM Corp. 1995

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

X

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
DataJoiner Implementation and Usage Guide
October 1995

Publication No. SG24-2566-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

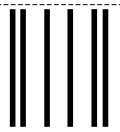
What other Technical Bulletins would you like to see published?

_____

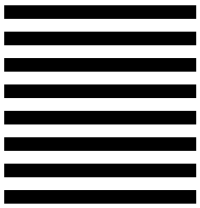
Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name	Address
Company or Organization	
Phone No.	

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 471, Building 070B
5600 COTTLE ROAD
SAN JOSE CA
USA 95193-0001

Fold and Tape **Please do not staple** Fold and Tape

Printed in U.S.A.