IBM DB2
Connectivity Supplement
Version 5

Document Number SDB2-CONN-SU

IBM DB2

Connectivity Supplement

Version 5

SDB2-CONN-SU

IBM DB2

Connectivity Supplement

Version 5

SDB2-CONN-SU

Before using this information and the product it supports, be sure to read the general information under Appendix A, “Notices” on page 137.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in U.S. or 1-800-IBM-4YOU in Canada.

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

Contents

Welcome to the DB2 Connectivity Supplement!

This book provides additional information to help you install and configure different DB2 RDBMS products as DRDA application requesters or as application servers. This information is provided to assist you in setting up:

- IBM DB2 Universal Database Version 5 servers, running as DRDA application servers (AS).
- IBM DB2 Connect (formerly DDCS) Version 5 application requesters (AR).
- Other DRDA-conformant products.

The information contained in this book is provided as a supplement to the information contained in the following manuals:

- *Quick Beginnings* for DB2 Universal Database Enterprise Edition Version 5
- *Quick Beginnings* for DB2 Connect Enterprise Edition Version 5
- *Quick Beginnings* for DB2 Connect Personal Edition Version 5.

For the latest and most up-to-date information on the host products (DB2 for OS/390, DB2 for AS/400, and DB2 for VSE & VM), you should refer to the documentation provided with them.

For information about configuring the LU 6.2 Syncpoint Manager (SPM) provided with DB2 Universal Database and DB2 Connect Enterprise Edition Version 5.0, please refer to *DB2 Connect Enterprise Edition Quick Beginnings*.

How This Book is Structured

This book is structured as follows:

Who Should Use This Book

This book is intended for anyone who has installed DB2 Universal Database or DB2 Connect and wants to know more about connectivity in the context of the subjects listed in the previous section.

Other Information Sources

This section lists other information sources which may be useful.

Using the World Wide Web

You can find the most recent information about DB2 Connect, DB2 Universal Database, and other IBM software products on the World Wide Web. This includes the latest publications, as well as technical hints and tips in the form of Technotes:

1. Set your Web browser to the following URL:

   `http://www.software.ibm.com/data/db2/library/`

2. Select “DB2 Universal Database”.
3. For example, search for “Technotes” using the keywords “DDCS”, “DRDA”, or “Connect”.

Related DRDA Publications

The following books contain related information and may be referenced in this manual.

Form number	Book title
SC26-4783	*Distributed Relational Database Architecture Connectivity Guide*
SC26-4773	*Distributed Relational Database Architecture Application Programming Guide*
SC26-4782	*Distributed Relational Database Architecture Problem Determination Guide*
SC26-4650	*Planning for Distributed Relational Database Architecture*
GC26-3195	*Distributed Relational Database Architecture Every Manager's Guide*
G321-5482	*IBM Distributed Data Management Architecture Level 3: Reference*

Related DRDA Server Publications

Related DRDA server publications include the following books from the DB2 for AS/400, DB2 for MVS/ESA, and DB2 for VSE & VM libraries.

Form number	Book title
SC41-5702	*AS/400 Distributed Database Programming*
SC41-9609	*AS/400 SAA Structured Query Language/400 Programmer's Guide*
SC41-9608	*AS/400 SAA Structured Query Language/400 Reference*
GC21-8180	*AS/400 Communications Configuration Reference*

Form number	Book title
SC26-8958	*DB2 for OS/390 Application Programming and SQL Reference*
SC26-8960	*DB2 for OS/390 Command Reference*
GC26-8970	*DB2 for OS/390 Installation Reference*
SC26-8964	*DB2 for OS/390 Reference for Remote DRDA Requesters and Servers*
SC26-8966	*DB2 for OS/390 SQL Reference*
SC26-8957	*DB2 for OS/390 Administration Guide*
SC26-8967	*DB2 for OS/390 Utility Guide and Reference*
SH09-8087	*DB2 for VSE & VM SQL Reference*
SC26-3255	*IBM SQL Reference*

Other Related Publications

Form number	Book title
SG24-2212	*DRDA Support for TCP/IP in DB2 for OS/390 V5.1 and DB2 Universal Database V5.0*
SC33-0814	*CICS for AIX Application Programming Guide*
SC33-0931	*CICS for AIX Customization and Operation Guide*
S10J-7888	*DB2 Connect Enterprise Edition Quick Beginnings*
S10J-8162	*DB2 Connect Personal Edition Quick Beginnings*
GG24-4308	*DDCS for OS/2 to DB2 Performance Benchmark*
GG24-3771	*Distributed Relational Database Architecture: Using OS/2 DRDA Client Support with DB2*
GG24-3926	*OS/2 DDCS and DB2 V2.3 Distributed relational Database Performance: Early Experience*
GG24-4155	*Distributed Relational Database Architecture: Using DDCS for AIX DRDA support with DB2 for MVS/ESA and DB2 for AS/400*
GG24-4311	*Distributed Relational Database Architecture Cross Platform Connectivity and Application*
SC23-2443	*Encina for AIX Product Family Overview*
SC31-8094	*SNA Server for AIX and SNA Server Gateway for AIX Performance Guide*

Chapter 1. Connecting DB2 for MVS/ESA in a DRDA Network

DB2 for MVS/ESA is the IBM relational database management system for MVS/XA, and MVS/ESA systems. DB2 for MVS/ESA Version 2 Release 3 was the first release of DB2 for MVS/ESA able to share distributed relational data with other DBMSs supporting DRDA protocols. This chapter describes how DB2 for MVS/ESA provides support for distributed relational database systems. If you are working with DB2 for OS/390 *do not use this chapter*: go instead to Chapter 2, "Connecting DB2 for OS/390 in a DRDA Network" on page 35.

The primary emphasis of the information in this chapter is on connecting unlike DRDA systems to DB2 for MVS/ESA on MVS systems. For information about connecting two DB2 for MVS/ESA systems, or more detailed information describing how to define DRDA connections to DB2 for MVS/ESA, see the discussion of connecting distributed database systems in the *IBM Database 2 Administration Guide*.

With the AnyNet Feature of VTAM Version 4 Release 2, you can run APPC over a TCP/IP network. The AnyNet Feature consists of AnyNet/MVS, which runs in a host, and AnyNet/2, which runs in a workstation and is downloaded from the host. Any APPC application is accessible to end users in a TCP/IP network without change to the application. Using APPC over TCP/IP, an application program on MVS/ESA can communicate with another APPC application program running with AnyNet APPC over TCP/IP on MVS/ESA, OS/2, AIX/6000, OS/400, or Windows. See *VTAM AnyNet Feature for V4R2 Guide to SNA over TCP/IP* for more information.

DB2 for MVS/ESA

Figure 1 shows an MVS system running a single copy of DB2 for MVS/ESA. It is also possible to run multiple copies of DB2 for MVS/ESA on a single MVS system. To identify copies of DB2 for MVS/ESA within a given MVS system (or copies of DB2 for MVS/ESA within an MVS/JES complex), each DB2 system is given a *subsystem name*, a one- to four- character string unique within an MVS/JES complex. In Figure 1, the DB2 for MVS/ESA subsystem name is *xxxx*. Three of the MVS address space names are prefixed by the DB2 for MVS/ESA subsystem name. These three address spaces make up the DB2 for MVS/ESA product.

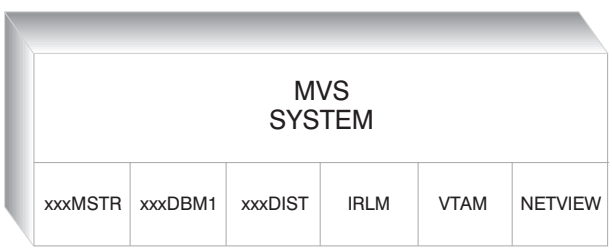

Figure 1. MVS Address Spaces used by DB2 for MVS/ESA

Figure 1 shows the MVS address spaces involved in distributed database processing with DB2 for MVS/ESA. These address spaces work together to allow DB2 for MVS/ESA users to access local relational databases and communicate with remote DRDA systems. The purpose of each address space is as follows:

xxxxMSTR The system services address space for the DB2 for MVS/ESA product responsible for starting and stopping DB2 for MVS/ESA, and controlling local access to DB2 for MVS/ESA.

xxxxDBM1 The database services address space responsible for accessing relational databases controlled by DB2 for MVS/ESA. This is where the input and output to database resources is performed on behalf of SQL application programs.

xxxxDIST The portion of DB2 for MVS/ESA that provides distributed database capabilities; also known as the *Distributed Data Facility* (DDF). When a distributed database request is received, DDF passes the request to *xxxx*DBM1, so that the required database I/O operations can be performed. This book describes DDF in detail.

IRLM The lock manager used by DB2 for MVS/ESA to control access to database resources.

VTAM The SNA Communications Manager for the MVS system. DDF uses VTAM to perform distributed database communications on behalf of DB2 for MVS/ESA.

NETVIEW The network management focal point product on MVS systems. When errors occur during distributed database processing, DDF records error information (also known as *alerts*) in the NetView hardware monitor database. System administrators can use NetView to examine the errors stored in the hardware monitor database, or provide automated command procedures to be invoked when alert conditions are recorded.

NetView can also be used to diagnose VTAM communication errors. For more information, see the *Distributed Relational Database Architecture Problem Determination Guide*.

Figure 1 on page 1 does not show any SQL application programs. When an application program uses DB2 to issue SQL statements, the application program must attach to the DB2 for MVS/ESA product in one of the following ways:

TSO Batch jobs and end users logged on to TSO are connected to DB2 for MVS/ESA through the TSO attach facility. This is the technique used to connect SPUFI and most QMF applications to DB2 for MVS/ESA.

CICS/ESA When a CICS/ESA application issues SQL calls, the CICS/ESA product uses the CICS attach interface to route SQL requests to DB2 for MVS/ESA.

IMS/ESA Transactions running under the control of IMS/ESA use the IMS attach interface to pass SQL statements to DB2 for MVS/ESA for processing.

DDF The Distributed Data Facility is responsible for connecting distributed applications to DB2 for MVS/ESA.

CAF The call attachment facility allows user-written subsystems to connect directly to DB2 for MVS/ESA.

DB2 for MVS/ESA Implementation

DRDA defines types of distributed database management system functions. DB2 for MVS/ESA V2R3 supports remote unit of work. With remote unit of work, an application program executing in one system can access data at a remote DBMS using the SQL provided by that remote DBMS. DB2 for MVS/ESA V3R1 supports distributed unit of work. With distributed unit of work, an application program executing in one system can access data at multiple remote DBMSs using SQL provided by remote DBMSs. For more information on the types of distribution defined by DRDA, see the *DRDA Connectivity Guide*.

As shown in Figure 2 on page 4, DB2 for MVS/ESA supports three configurations of distributed database connections using two access methods:

[1] *System-directed access* allows a DB2 for MVS/ESA requester to connect to one or more DB2 for MVS/ESA servers. The connection established between the DB2 for MVS/ESA requester and server does not adhere to the protocols defined in DRDA and cannot be used to connect non-DB2 for MVS/ESA products to DB2 for MVS/ESA. This type of connection is established by coding three-part names or aliases in the application.

[2] *Application-directed access* allows a DB2 for MVS/ESA or non-DB2 for MVS/ESA requester to connect to one or more DB2 for MVS/ESA or non-DB2 for MVS/ESA application servers using DRDA protocols. The number of application servers that can be connected to the application requester at one time depends on the level of DB2 for MVS/ESA of the application requester. If the application requester is DB2 for MVS/ESA V2R3, then only one application server can be connected at a time. This type of connection is established by coding SQL CONNECT statements in the application. If the application requester is DB2 for MVS/ESA V3R1 , then one or more application servers can be connected at a time.

[3] Application-directed and system-directed access can be used together to establish connections.

The term *secondary server* describes systems acting as servers to the application server.

If all systems in a configuration support two-phase commit, then distributed unit of work (multiple-site read and multiple-site update) is supported. If not all systems support two-phase commit, updates within a unit of work are either restricted to a single site that does not support two-phase commit, or to the subset of sites that support two-phase commit.

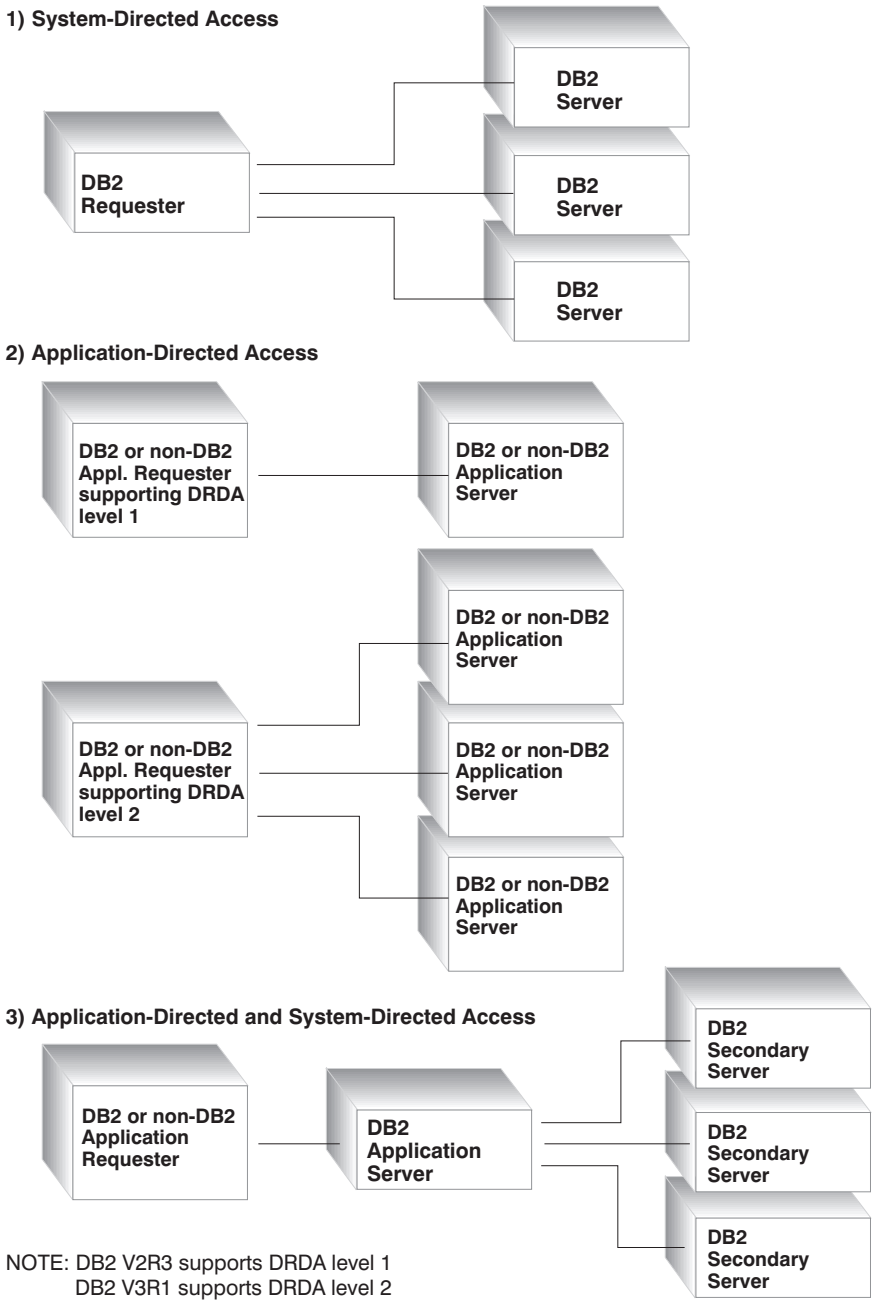

Figure 2. DB2 for MVS/ESA Distributed Connections

Table 1 on page 5 compares the DB2 for MVS/ESA distributed database connection types.

Table 1. Comparison of DB2 for MVS/ESA Distributed Database Connections

[1] System-Directed Access	**[2] Application-Directed Access (with all systems having two-phase commit)**	**[3] Application-Directed and System-Directed Accesses**
All partners must be DB2 for MVS/ESA systems	Can interconnect any two DRDA systems	Application requester can be any DRDA system; servers must be DB2 for MVS/ESA systems
Can connect directly to many partners	Can connect directly to many partners	Application requester connects directly to Application Servers; Application Servers can connect to many DB2 for MVS/ESA secondary servers
Each SQL application can have multiple APPC conversations with each server	Each SQL application has one APPC conversation with each server	SQL application has one APPC conversation with each server; DB2 for MVS/ESA application server can establish many APPC conversations to each server for the application
Can access both local and remote resources in one commit scope	Can access both local and remote resources in one commit scope	Application requester and Application Server can access local and remote data
More efficient at large queries and multiple concurrent queries	More efficient at SQL statements that are executed very few times in one commit scope	Application requester-Application Server connection behaves like **[2]**; secondary server connections behave like **[1]**
Can support static or dynamic SQL, but server dynamically binds static SQL the first time it is executed in a commit scope	Can issue static or dynamic SQL	Application requester and Application Server can issue static or dynamic SQL; secondary servers support static or dynamic SQL, but dynamically bind static SQL the first time it is executed in a commit scope
Limited to SQL INSERT, DELETE, and UPDATE statements, and to statements that support SELECT	Can use any statement supported by the system that executes the statement	Application servers supports any SQL; secondary servers support only DML SQL (for example, CREATE or ALTER)

Setting Up the Application Requester

DB2 for MVS/ESA implements the DRDA application requester support as an integral part of the DB2 for MVS/ESA Distributed Data Facility (DDF). DDF can be stopped independently from the local DB2 for MVS/ESA database management facilities, but it cannot run in the absence of the local DB2 for MVS/ESA database management support.

When DB2 for MVS/ESA acts as an Application Requester, it can connect to a DB2 for MVS/ESA Application Server or any other product that supports the DRDA architecture.

For the DB2 for MVS/ESA Application Requester to provide distributed database access, you need to do the following:

- "Provide Network Information"—The Application Requester must be able to accept RDB_NAME values and translate these values into SNA NETID.LUNAME values. DB2 for MVS/ESA uses the *DB2 for MVS/ESA communications database* to register RDB_NAMEs and their corresponding network parameters. The communications database allows the DB2 for MVS/ESA Application Requester to pass the required SNA information to VTAM when making distributed database requests.
- "Provide Security" on page 16— For remote database requests to be accepted by the Application Server, the Application Requester must provide the security information required by the server. DB2 for MVS/ESA uses the communications database and RACF to provide the required network security information.
- "Represent Data" on page 21—You must ensure that the CCSID of the application requester is compatible with the application server.

Provide Network Information

Much of the processing in a distributed database environment requires exchanging messages with other locations in your network. For this processing to be performed correctly, you need to do the following:

1. Define the local system
2. Define the remote systems
3. Define the communications
4. Set RU sizes and pacing

Defining the Local System

Each program in the network is assigned a NETID and an LU name, so your DB2 for MVS/ESA Application Requester must have a NETID.LUNAME value when it connects to the network. Because the DB2 for MVS/ESA Application Requester is integrated into the local DB2 for MVS/ESA database management system, the Application Requester must also have an RDB_NAME. In the DB2 for MVS/ESA publications, DB2 for MVS/ESA refers to the RDB_NAME as a *location* name.

Define the DB2 for MVS/ESA Application Requester to the SNA network as follows:

1. Select an LU name for your DB2 for MVS/ESA system. The NETID for your DB2 for MVS/ESA system is automatically obtained from VTAM when DDF starts.
2. Define the LU name and location name in the DB2 for MVS/ESA *bootstrap data set* (BSDS). (DB2 for MVS/ESA restricts the location to 16 characters.)
3. Create a VTAM APPL definition to register the selected LU name with VTAM.

Configuring the DDF BSDS: DB2 for MVS/ESA reads the BSDS during startup processing to obtain system installation parameters. One of the records stored in the BSDS is called the *DDF record*, because it contains the information used by DDF to connect to VTAM. This information consists of the following:

- The location name for the DB2 for MVS/ESA system
- The LU name for the DB2 for MVS/ESA system
- The password used when connecting the DB2 for MVS/ESA system to VTAM

You can supply the DDF BSDS information to DB2 for MVS/ESA in two ways:

- Use the DDF installation panel DSNTIPR when you first install DB2 for MVS/ESA to provide the required DDF BSDS information. Many of the install parameters are not discussed here because it is more important to know how to connect DB2 for MVS/ESA to VTAM. Figure 3 shows how to use the installation panel to record location name SYDNEY, the LU name LUDBD1, and password PSWDBD1 in the DB2 for MVS/ESA BSDS.

```
 1 DDF STARTUP OPTION   ===> AUTO      NO (DDF not startable),
                                       AUTO (automatic start up), or
                                       COMMAND (start by command)
 2 DB2 LOCATION NAME    ===> SYDNEY    The name other DB2s use to
                                       refer to this DB2
 3 DB2 NETWORK LUNAME   ===> LUDBD1    The name VTAM uses to refer to this DB2
 4 DB2 NETWORK PASSWORD ===> PSWDBD1   Password for connecting to other DB2s
 5 RLST ACCESS ERROR    ===> NOLIMIT   Action on non-local RLST access error
                                       NOLIMIT   - Run without limit
                                       NORUN     - Do not run at all
                                       1-5000000 - Limit in CPU service units
 6 SAMPLE TEST LOCATION ===> DALLAS            Location name used in phase 6
                                               (DSNTEJ6) of DB2 sample
                                               applications
PRESS:  ENTER to continue   END to exit   HELP for more information
```

Figure 3. DB2 for MVS/ESA Installation Panel DSNTIPR

- If DB2 for MVS/ESA is already installed, you can use the change log inventory utility (DSNJU003) to update the information in the BSDS.

  Figure 4 on page 8 shows how to update the BSDS with location name SYDNEY, the LU name LUDBD1, and password PSWDBD1.

```
//SYSADMB JOB ,'DB2 2.3 JOB',CLASS=A
//*
//*        CHANGE LOG INVENTORY:
//*        UPDATE BSDS WITH
//*             - DB2 LOCATION NAME FOR SYDNEY
//*             - VTAM LUNAME (LUDBD1)
//*             - DB2/VTAM PASSWORD
//*
//DSNBSDS EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DSN230.DSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DSNC230.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DSNC230.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 DDF    LOCATION=SYDNEY,LUNAME=LUDBD1,PASSWORD=PSWDBD1
//*
```

Figure 4. Sample Bootstrap Data Set DDF Definition

When DDF is started (either automatically at DB2 for MVS/ESA startup or by the DB2 for MVS/ESA START DDF command), it connects to VTAM, passing the LU name and password to VTAM. VTAM recognizes the DB2 for MVS/ESA system by checking the LU name and password (if a VTAM password is required) with the values defined in the DB2 for MVS/ESA VTAM APPL statement. The VTAM password is used to verify that DB2 for MVS/ESA is authorized to use the specified LU name on the VTAM system. The VTAM password is not transmitted through the network, and it is not used to connect other systems in the network to DB2 for MVS/ESA.

If VTAM does not require a password, omit the PASSWORD= keyword on the change log inventory utility. The absence of the keyword indicates that no VTAM password is required.

Creating a VTAM APPL definition: After you define the VTAM LU name and password to DB2 for MVS/ESA, you need to register these values with VTAM. VTAM uses the APPL statement to define local LU names. Figure 5 on page 9 shows how to define the LU name LUDBD1 to VTAM.

```
DB2APPLS VBUILD TYPE=APPL
*
*--------------------------------------------------------------------*
*                                                                    *
*           APPL DEFINITION FOR THE SYDNEY DB2 SYSTEM                *
*                                                                    *
*--------------------------------------------------------------------*
*
LUDBD1   APPL  APPC=YES,                                            X
               AUTH=(ACQ),                                          X
               AUTOSES=1,                                           X
               DMINWNL=10,                                          X
               DMINWNR=10,                                          X
               DSESLIM=20,                                          X
               EAS=9999,                                            X
               MODETAB=RDBMODES,                                    X
               PRTCT=PSWDBD1,                                       X
               SECACPT=ALREADYV,                                    X
               SRBEXIT=YES,                                         X
               VERIFY=NONE,                                         X
               VPACING=2,                                           X
               SYNCLVL=SYNCPT,                                      X
               ATNLOSS=ALL                                          X
```

Figure 5. Sample DB2 for MVS/ESA APPL Definition

Many keywords are available on the VTAM APPL statement. The meaning of the keywords is discussed in detail in the *DB2 Administration Guide*. The only keywords discussed here address topics in this book. The keywords of interest in Figure 5 are described as follows:

LUDBD1

VTAM uses the APPL statement label as the LU name. In this case, the LU name is LUDBD1. The APPL syntax does not allow room for a complete NETID.LUNAME value. The NETID value is not specified on the VTAM APPL statement, because all VTAM applications are automatically assigned the NETID for the VTAM system.

AUTOSES=1

The number of SNA contention winner sessions that start automatically when an APPC Change Number of Sessions (CNOS) request is issued. A nonzero value must be supplied with AUTOSES to inform DB2 for MVS/ESA in all cases when VTAM CNOS processing fails.

You do not have to automatically start all the APPC sessions between any two distributed database partners. If the AUTOSES value is less than the contention winner limit (DMINWNL), VTAM delays starting the remaining SNA sessions until they are required by a distributed database application.

DMINWNL=10

The number of sessions on which this DB2 for MVS/ESA system is the contention winner. The DMINWNL parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.SYSLUMODES table in the DB2 for MVS/ESA communications database.

DMINWNR=10

The number of sessions on which the partner system is the contention winner. The DMINWNR parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.SYSLUMODES table in the DB2 for MVS/ESA communications database.

DSESLIM=20

The total number of sessions (winner and loser sessions) you can establish between DB2 for MVS/ESA and another distributed system for a specific mode group name. The DSESLIM parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.SYSLUMODES table in the DB2 for MVS/ESA communications database.

If the partner cannot support the number of sessions requested on the DSESLIM, DMINWNL, or DMINWNR parameters, the CNOS process negotiates new values for these parameters that are acceptable to the partner.

EAS=9999

An estimate of the total number of sessions that this VTAM LU requires.

MODETAB=RDBMODES

Identifies the VTAM MODE table where each DB2 for MVS/ESA mode name exists.

PRTCT=PSWDBD1

Identifies the VTAM password to use when DB2 for MVS/ESA attempts to connect to VTAM. If the PRTCT keyword is omitted, no password is required, and you should omit the PASSWORD= keyword from the DB2 for MVS/ESA change log inventory utility.

SECACPT=ALREADYV

Identifies the highest SNA conversation-level security value accepted by this DB2 for MVS/ESA system when it receives a distributed database request from a remote system. The ALREADYV keyword indicates this DB2 for MVS/ESA system can accept three SNA session security options from other DRDA systems that request data from this DB2 for MVS/ESA system:

- SECURITY=SAME (an already-verified request that contains only the requester's user ID).
- SECURITY=PGM (a request containing the requester's user ID and password).
- SECURITY=NONE (a request containing no security information). DB2 for MVS/ESA rejects DRDA requests that specify SECURITY=NONE.

It is best to always specify SECACPT=ALREADYV, because the SNA conversation security level for each DB2 for MVS/ESA partner is taken from the DB2 for MVS/ESA communications database (the USERSECURITY column of the SYSIBM.SYSLUNAMES table). SECACPT=ALREADYV gives you the most flexibility in selecting values for USERSECURITY.

VERIFY=NONE
Identifies the level of SNA session security (partner LU verification) required by this DB2 for MVS/ESA system. The NONE value indicates that partner LU verification is not required.

DB2 for MVS/ESA does not restrict your choice for the VERIFY keyword. In a nontrusted network, VERIFY=REQUIRED is recommended. VERIFY=REQUIRED causes VTAM to reject partners that cannot perform partner LU verification. If you choose VERIFY=OPTIONAL, VTAM performs partner LU verification only for those partners that provide the support.

VPACING=2
Sets the VTAM pacing count to 2.

SYNCLVL=SYNCPT
Indicates that DB2 for MVS/ESA is able to support two-phase commit. VTAM uses this information to inform the partner that two-phase commit is available. If this keyword is present, DB2 for MVS/ESA automatically uses two-phase commit if the partner can support it.

ATNLOSS=ALL
Indicates that DB2 for MVS/ESA needs to be informed each time a VTAM session ends. This ensures that DB2 for MVS/ESA performs SNA resynchronization when required.

DSESLIM, DMINWNL, and DMINWNR allow you to establish default VTAM session limits for all partners. For partners that have special session limit requirements, the SYSIBM.SYSLUMODES table can be used to override the default session limits. For example, you might want to specify VTAM default session limits that are appropriate for your OS/2 systems. For other partners, you can create rows in the SYSIBM.SYSLUMODES table to define the desired session limits. Consider these sample values:

```
DSESLIM=4,DMINWNL=0,DMINWNR=4
```

These parameters allow each partner to create up to four sessions with DB2 for MVS/ESA, where the partner is the contention winner on each of the sessions. Because OS/2 creates the LU 6.2 conversations with DB2 for MVS/ESA, by making OS/2 the contention winner on the sessions, you gain a small performance advantage. If OS/2 has an available contention winner session, it does not have to ask for permission to start a new LU 6.2 conversation.

Defining the Remote Systems

When a DB2 for MVS/ESA application requests data from a remote system, DB2 for MVS/ESA searches the communications database tables to find information about the remote system, including a search on:

- The LU name and TPN
- The network security information required by the remote site
- The session limits and mode names used to communicate with the remote site

The communications database is a group of SQL tables managed by the DB2 for MVS/ESA system administrator. As the DB2 for MVS/ESA system administrator, you must use SQL to insert rows in the communications database to describe each potential DRDA partner. The communications database consists of five tables:

1. **SYSIBM.SYSLOCATIONS**

   This table allows DB2 for MVS/ESA to determine the LU name and TPN value for each RDB_NAME selected by a DB2 for MVS/ESA application. The columns are:

   LOCATION The RDB_NAME of the remote system. DB2 for MVS/ESA limits the RDB_NAME value to 16 bytes, which is two bytes shorter than the 18-byte limit defined in DRDA.

   LOCTYPE Currently not used; it must be blank.

   LINKNAME The LU name of the remote system.

   LINKATTR The TPN of the remote system. If the remote system is a DB2 for MVS/ESA system or the remote system uses the default DRDA TPN value (X'07F6C4C2'[1]), an empty string can be used to specify the TPN because DB2 for MVS/ESA automatically chooses the correct value.

   If the remote system requires a TPN value other than the default TPN value, you must supply this value here.

2. **SYSIBM.SYSLUNAMES**

   This table defines the network attributes of the remote systems. The columns are:

   LUNAME The LU name of the remote system.

   SYSMODENAME The VTAM logon mode name used to establish the DB2 for MVS/ESA-to-DB2 for MVS/ESA *intersystem* conversations for the DB2 for MVS/ESA secondary server support (system-directed access). A blank value in this column indicates IBMDB2LM should be used for DB2 for MVS/ESA system conversations.

   USERSECURITY The network security acceptance options required of the remote system when this DB2 for MVS/ESA system acts as a server for the remote system (*inbound security* requirements).

   ENCRYPTPSWDS Whether passwords exchanged with this partner are encrypted. Encrypted passwords are only supported by DB2 for MVS/ESA requesters and servers.

---

1 This TPN value *currently* applies to DB2 for VM .

MODESELECT Determines whether the SYSIBM.SYSMODESELECT table is used to select a VTAM logon mode entry (mode name) based on the end user and application making the request. If this column contains a 'Y', the SYSIBM.SYSMODESELECT table is used to obtain the mode name for each outbound distributed database request.

If MODESELECT contains anything other than a 'Y', the mode name IBMDB2LM is used for system-directed access requests, and the mode name IBMRDB is used for DRDA requests.

The MODESELECT column allows you to prioritize distributed database requests by specifying a VTAM class of service (COS) associated with the mode name.

USERNAMES The level of come-from checking and user ID translation required. This column also specifies the security parameters this DB2 for MVS/ESA subsystem uses when requesting data from the remote partner (*outbound security* requirements). USERNAMES can have the value I, O, or B.

3. **SYSIBM.SYSLUMODES**

   This table is used to define LU 6.2 session limits (CNOS limits) for each partner system. The columns are:

   LUNAME The LU name of the remote system.

   MODENAME The name of the VTAM logon mode whose limits are being specified. A blank value in the MODENAME column defaults to IBMDB2LM.

   CONVLIMIT The maximum number of active conversations between the local DB2 for MVS/ESA and the remote system for this logon mode. This value is used to override the DSESLIM parameter in the VTAM APPL definition statement for this logon mode, which supplies the default VTAM session limits for DB2 for MVS/ESA.

   The value selected in CONVLIMIT is used during CNOS to set the DMINWNR and DMINWNL values to CONVLIMIT/2.

   AUTO Whether CNOS processing and preallocation of sessions are initiated automatically at DDF startup or deferred until the first reference to the LU name via this logon mode.

4. **SYSIBM.SYSMODESELECT**

   This table allows you to specify different mode names for individual end users and DB2 for MVS/ESA applications. Because each VTAM mode name can have an associated class of service (COS), you can use this table to assign network transmission priorities to distributed database applications based on a combination of AUTHID, PLANNAME, and LUNAME. The columns are:

AUTHID The DB2 for MVS/ESA user's authorization ID (user ID). The default is blank, indicating the specified logon mode name applies to all authorization IDs.

PLANNAME The plan name associated with the application requesting access to a remote database system. The default is blank, indicating that the specified logon mode name applies to all plan names. The plan name used for the BIND PACKAGE command is DSNBIND.

LUNAME The LU name associated with the remote database system.

MODENAME The name of the VTAM logon mode to use when routing a distributed database request to the indicated remote system. The default is blank, indicating that IBMDB2LM should be used for system-directed access conversations and IBMRDB should be used for DRDA conversations.

5. **SYSIBM.SYSUSERNAMES**

   This table is used to manage end user names by providing passwords, name translations, and come-from checking. DB2 for MVS/ESA refers to the end user's name as an authorization ID. Most other products refer to this name as a user ID.

   With this table, you can use name translation to force different values to be used for the SNA user ID and the DB2 for MVS/ESA authorization ID. The name translation process is allowed for requests to a remote system (*outbound* requests) and for requests coming from a remote system (*inbound* requests). If passwords are not encrypted, this table is the source of the end user's password when both user ID and password are sent to a remote site. The columns are:

   TYPE The description of how the row is to be used (whether it is a row describing name translations for outbound or inbound/come-from checking requests).

   AUTHID For outbound name translation, this is the DB2 for MVS/ESA authorization ID to translate. For inbound name translation, this is the SNA user ID to translate. In either case, a blank AUTHID value applies to all authorization IDs or user IDs.

   LUNAME The LU name of the remote system to which this row applies. If blank, the NEWAUTHID value applies to all systems.

   NEWAUTHID The new end user name (either SNA user ID or DB2 for MVS/ESA authorization ID). Blank specifies that you do not need to translate the ID.

   PASSWORD The password used on the allocate conversation, if passwords are not encrypted (ENCRYPTPSWDS = 'N' in SYSIBM.SYSLUNAMES). If passwords are encrypted, this column is ignored.

Defining Communications

VTAM is the Communications Manager for MVS systems. VTAM accepts LU 6.2 verbs from DB2 for MVS/ESA and converts these verbs into LU 6.2 data streams you can transmit over the network. For VTAM to communicate with the partner applications defined in the DB2 for MVS/ESA communications database, you need to provide VTAM with the following information:

- The LU name for each server.

  When DB2 for MVS/ESA communicates with VTAM, DB2 for MVS/ESA is allowed to pass only an LU name (not NETID.LUNAME) to VTAM to identify the desired destination. This LU name must be unique within the LU names known by the local VTAM system, allowing VTAM to determine both the NETID and LU name from the LU name value passed by DB2 for MVS/ESA. When LU names are unique throughout an enterprise's SNA network, it greatly simplifies the VTAM resource definition process. However, this might not always be possible. If LU names within your SNA networks are not unique, you must use VTAM LU name translation to build the correct NETID.LUNAME combination for a nonunique LU name. This process is described in “Resource Name Translation” in the *VTAM Network Implementation Guide*.

  The placement and syntax of the VTAM definitions used to define remote LU names are highly dependent on how the remote system is logically and physically connected to the local VTAM system.

- The RU size, pacing window size, and class of service for each mode name. Create an entry in the VTAM mode table for each mode name specified in the communications database. You also need to define IBMRDB and IBMDB2LM.

- The VTAM and RACF profiles for the LU verification algorithm, if you intend to use partner LU verification.

Setting RU Sizes and Pacing

The VTAM mode table entries you define specify RU sizes and pacing counts. Failure to define these values correctly can have a negative impact on all VTAM applications.

After choosing RU sizes, session limits, and pacing counts, it is extremely important to consider the impact these values can have on the existing VTAM network. You should review the following items when you install a new distributed database system:

- For VTAM CTC connections, verify that the MAXBFRU parameter is large enough to handle your RU size plus the 29 bytes VTAM adds for the SNA request header and transmission header. MAXBFRU is measured in units of 4K bytes, so MAXBFRU must be at least 2 to accommodate a 4K RU.

- For NCP connections, make sure that MAXDATA is large enough to handle your RU size plus 29 bytes. If you specify an RU size of 4K, MAXDATA must be at least 4125.

  If you specify the NCP MAXBFRU parameter, select a value that can accommodate the RU size plus 29 bytes. For NCP, the MAXBFRU parameter defines the number of VTAM I/O buffers that can be used to hold the PIU. If you choose an

IOBUF buffer size of 441, MAXBFRU=10 processes a 4K RU correctly because 10*441 is greater than 4096+29.

- *DRDA Connectivity Guide* describes how to assess the impact your distributed database has on the VTAM IOBUF pool. If you use too much of the IOBUF pool resource, VTAM performance is degraded for all VTAM applications.

Provide Security

When a remote system performs distributed database processing on behalf of an SQL application, it must be able to satisfy the security requirements of the Application Requester, the Application Server, and the network connecting them. These requirements fall into one or more of the following categories:

- Selection of end user names
- Network security parameters
- Database manager security
- Security enforced by an external security subsystem
- Data representation

Selecting End User Names

On MVS systems, end users are assigned a 1 to 8-character *user ID*. This user ID value must be unique within a particular MVS system, but might not be unique throughout the SNA network. For example, there can be a user named JONES on the NEWYORK system, and another user named JONES on the DALLAS system. If these two users are the same person, no conflict exists. However, if the JONES in DALLAS is a different person than the JONES in NEWYORK, the SNA network (and consequently the distributed database systems within that network) cannot distinguish between JONES in NEWYORK and JONES in DALLAS. If you do not correct this situation, JONES in DALLAS can use the privileges granted to JONES at the NEWYORK system.

To eliminate naming conflicts, DB2 for MVS/ESA provides support for end user name translation. When an application at the DB2 for MVS/ESA Application Requester makes a distributed database request, DB2 for MVS/ESA performs name translation if the communications database specifies that *outbound name translation* is required. If outbound name translation is selected, DB2 for MVS/ESA always forces a password to be sent with each outbound distributed database request.

Outbound name translation in DB2 for MVS/ESA is activated by setting the USERNAMES column in the SYSIBM.SYSLUNAMES table to either `'O'` or `'B'`. If USERNAMES is set to `'O'`, end user name translation is performed for outbound requests. If USERNAMES is set to `'B'`, end user name translation is performed for both inbound and outbound requests.

Because DB2 for MVS/ESA authorization is dependent on both the end user's user ID and the user ID of the DB2 for MVS/ESA plan or package owner, the end user name translation process is performed for the end user's user ID, the plan owner's user ID,

and the package owner's user ID.[2] The name translation process searches the SYSIBM.SYSUSERNAMES table in the following sequence to find a row that matches one of the following patterns (TYPE.AUTHID.LUNAME):

1. O.AUTHID.LUNAME—A translation rule for a specific end user to a specific partner system.
2. O.AUTHID.blank—A translation rule for a specific end user to any partner system.
3. O.blank.LUNAME—A translation rule for any user to a specific partner system.

If no matching row is found, DB2 for MVS/ESA rejects the distributed database request. If a row is found, the value in the NEWAUTHID column is used as the authorization ID. (A blank NEWAUTHID value indicates the original name is used without translation.)

Consider the example discussed earlier. You want to give JONES in NEWYORK a different name (NYJONES) when JONES makes distributed database requests to DALLAS. In the example, assume that the application used by JONES is owned by DSNPLAN (the DB2 for MVS/ESA plan owner), and you do not need to translate this user ID when it is sent to DALLAS. The SQL statements required to supply the name translation rules in the communications database are shown in Figure 6.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', 'O');
INSERT INTO SYSIBM.SYSLOCATIONS
     (LOCATION, LOCTYPE, LINKNAME, LINKATTR)
  VALUES ('DALLAS', ' ', 'LUDALLAS', '');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'JONES', 'LUDALLAS', 'NYJONES', 'JONESPWD');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'DSNPLAN', 'LUDALLAS', ' ', 'PLANPWD');
```

Figure 6. SQL for Outbound Name Translation

The resulting communications database tables are shown in Figure 7 on page 18:

2 If the request is being sent to a DB2 for MVS/ESA server, name translation is also performed for the package owner and plan owner. Package and plan owner names never have passwords associated with them.

NEWYORK.SYSIBM.SYSLOCATIONS

LOCATION	LOCTYPE	LINKNAME	LINKATTR
DALLAS		LUDALLAS	

NEWYORK.SYSIBM.SYSLUNAMES

LUNAME	SYSMODENAME	USERSECURITY	ENCRYPTPSWDS	MODESELECT	USERNAMES
LUDALLAS		A	N	N	O

NEWYORK.SYSIBM.SYSUSERNAMES

TYPE	AUTHID	LUNAME	NEWAUTHID	PASSWORD
0 0	JONES DSNPLAN	LUDALLAS LUDALLAS	NYJONES	JONESPWD PLANPWD

Figure 7. Outbound Name Translation

Network Security

After the Application Requester selects the end user names to represent the remote application, the Application Requester must provide the required LU 6.2 network security information. LU 6.2 provides three major network security features:

- Session-level security, which is controlled by the VERIFY keyword on the VTAM APPL statement. See the discussion following Figure 5 on page 9 for a description of how to specify session-level security options.
- Conversation-level security, which is controlled by the contents of the SYSIBM.SYSLUNAMES table.
- Data encryption, which is supported only for VTAM 3.4 and later releases of VTAM.

Because the Application Server is responsible for managing the database resources, the Application Server dictates which network security features are required of the Application Requester. You must record the conversation-level security requirements of

each Application Server in the SYSIBM.SYSLUNAMES table by setting the USERNAMES column of the SYSIBM.SYSLUNAMES table to reflect the application server's requirement.

The possible SNA conversation security options are:

SECURITY=SAME

This is also known as already-verified security because only the end user's user ID is sent to the remote system (no password is transmitted). Use this level of conversation security when the USERNAMES column in SYSIBM.SYSLUNAMES does not contain 'O' or 'B'.

Because DB2 for MVS/ESA ties end user name translation to outbound conversation security, it does not allow you to use SECURITY=SAME when outbound end user name translation is activated.

SECURITY=PGM

This causes the end user's ID and password to be sent to the remote system for validation. Use this security option when the USERNAMES column of the SYSIBM.SYSLUNAMES table contains either an 'O' or 'B'.

Depending upon options specified in the SYSIBM.SYSLUNAMES table, DB2 for MVS/ESA obtains the end user's password from two different sources:

- Unencrypted passwords are obtained from the PASSWORD column of the SYSIBM.SYSUSERNAMES table. DB2 for MVS/ESA extracts passwords from the SYSIBM.SYSUSERNAMES table when the ENCRYPTPSWDS column in SYSIBM.SYSLUNAMES is not set to 'Y'. Passwords obtained from this source can be transmitted to any DRDA Application Server.

  Figure 8 defines passwords for SMITH and JONES. The LUNAME column in the example contains blanks, so these passwords are used for any remote system SMITH or JONES attempts to access.

```
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'JONES', ' ', ' ', 'JONESPWD');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'SMITH', ' ', ' ', 'SMITHPWD');
```

Figure 8. Sending Passwords to Remote Sites

- Encrypted passwords are sent to the remote site when the ENCRYPTPSWDS column of SYSIBM.SYSLUNAMES contains 'Y'. Encrypted passwords are extracted from RACF (or a RACF-equivalent product), and can only be interpreted by another DB2 for MVS/ESA system. When communicating with a non-DB2 for MVS/ESA system, do not set ENCRYPTPSWDS to 'Y'.

  DB2 for MVS/ESA searches the SYSIBM.SYSUSERNAMES table to determine the user ID (NEWAUTHID value) to transmit to the remote system. This

translated name is used for the RACF password extraction. If you do not want to translate names, you must create rows in SYSIBM.SYSUSERNAMES that cause names to be sent without translation. Figure 9 on page 20 allows requests to be sent to LUDALLAS and LUNYC without translating the end user's name (user ID).

```
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', ' ', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', ' ', 'LUDALLAS', ' ', ' ');
```

Figure 9. Sending Encrypted Passwords to Remote Sites

SECURITY=NONE

This option is not supported by DRDA, so DB2 for MVS/ESA has no provision for this security option.

Database Manager Security

One way the Application Requester can participate in distributed database security is through outbound name translation, as stated earlier in “Selecting End User Names” on page 16. You can use outbound name translation to control access to each Application Server, based on the identity of the end user making the request and the application making the request. Other ways the DB2 for MVS/ESA Application Requester contributes to the distributed system security are:

Binding remote applications

End users bind remote applications at the Application Server with the DB2 for MVS/ESA BIND PACKAGE command. DB2 for MVS/ESA *does not* restrict the use of the BIND PACKAGE command at the requester. However, an end user cannot use a remote package until the package is included in a DB2 for MVS/ESA plan. DB2 for MVS/ESA *does* restrict the use of the BIND PLAN command. An end user cannot add the remote package to a plan unless the end user is given either the BIND or BINDADD privilege with the DB2 for MVS/ESA GRANT statement.

When you bind a package, use the ENABLE/DISABLE option to specify whether the package is to be used by TSO, CICS/ESA, IMS/ESA, or a remote DB2 for MVS/ESA subsystem.

Executing remote applications

For the DB2 for MVS/ESA end user to run a remote application, the end user must have authority to run the DB2 for MVS/ESA plan associated with that application. The DB2 for MVS/ESA plan owner automatically has authority to run the plan. Other end users can be given authority to run the plan with the DB2 for MVS/ESA GRANT EXECUTE statement. In this way, the owner of a distributed database application can control use of the application on a user-by-user basis.

Security Subsystem

The external security subsystem on MVS systems is provided by RACF and other products that provide an interface compatible with RACF. The DB2 for MVS/ESA Application Requester does not have any direct calls to the external security subsystem, with the exception of the encrypted password support described in “Network Security” on page 18. However, the external security subsystem is used indirectly at the Application Requester in the following situations:

- The product responsible for attaching the end user to DB2 for MVS/ESA uses the external security subsystem to validate the end user's identity (user ID and password). This occurs before the end user is attached to DB2 for MVS/ESA. As stated earlier, CICS/ESA, TSO, and IMS/ESA are examples of products that attach end users to DB2 for MVS/ESA.
- If you use SNA session-level security (via the VERIFY keyword on the DB2 for MVS/ESA VTAM APPL statement), the external security subsystem is invoked by VTAM to validate the identity of the remote system.

Represent Data

DB2 for MVS/ESA is shipped with a default installation coded character set identifier (CCSID) of 500. This default is probably *not* correct for your installation.

When installing DB2 for MVS/ESA, you must set the installation CCSID to the CCSID of the characters generated and sent to DB2 for MVS/ESA by the input devices at your site. This CCSID is generally determined by the national language you use. If the installation CCSID is not correct, character conversion will produce incorrect results. See *DB2 Connect User's Guide* for a list of the supported CCSIDs for each country or national language.

You must ensure that your DB2 for MVS/ESA subsystem has the ability to convert from each application server's CCSID to your DB2 for MVS/ESA subsystem's installation CCSID. DB2 for MVS/ESA provides conversion tables for the most common combinations of source and target CCSIDs, but not for every possible combination. You can add to the set of available conversion tables and conversion routines if you need to. See the *DB2 Administration Guide*, for more information about DB2 for MVS/ESA character conversion.

Setting Up the Application Server

The application server support in DB2 for MVS/ESA allows DB2 for MVS/ESA to act as a server for DRDA application requesters. The Application Requester connected to a DB2 for MVS/ESA Application Server can be:

- A DB2 for MVS/ESA requester
- A DB2 Universal Database requester, which can run on AIX, HP-UX, OS/2, Solaris, Windows 3.1, Windows 3.11 for Workgroups, Windows 95, or Windows NT.
- A DDCS Version 2 requester, which can run on AIX, HP-UX, OS/2, Solaris, Windows 3.1, Windows 3.11 for Workgroups, Windows 95, or Windows NT, as well as SCO, SGI, or SINIX.

- A DB2 Connect Version 5 requester, which can run on AIX, HP-UX, OS/2, Solaris, Windows 3.1, Windows 3.11 for Workgroups, Windows 95, or Windows NT.
- An OS/400 requester
- A DB2 for VM requester
- Any other product that supports the DRDA Application Requester protocols

For any Application Requester connected to a DB2 for MVS/ESA Application Server, the DB2 for MVS/ESA Application Server supports database access as follows:

- The Application Requester is permitted to access tables stored at the DB2 for MVS/ESA application server. The Application Requester must create a package at the DB2 for MVS/ESA Application Server before the application can be run. The DB2 for MVS/ESA Application Server uses the package to locate the application's SQL statements at execution time.
- The Application Requester can inform the DB2 for MVS/ESA Application Server that access must be restricted to read-only activities if the DRDA requester-server connection does not support the two-phase commit process. For example, a DB2 for MVS/ESA V2R3 requester with a CICS front end would inform the DB2 for MVS/ESA application server that updates are not allowed.
- The Application Requester can also be permitted to access tables stored at other DB2 for MVS/ESA systems in the network using system-directed access. System-directed access allows the application requester to establish connections to multiple database systems in a single unit of work.

Provide Network Information

For the DB2 for MVS/ESA Application Server to properly process distributed database requests, you must take the following steps:

1. Define the application server to the local Communications Manager.
2. Define each potential secondary server destination so the DB2 for MVS/ESA application server can reroute SQL requests to their final destination.
3. Provide the necessary security.
4. Provide for data representation.

Defining the Application Server

For the Application Server to receive distributed database requests, it must be defined to the local Communications Manager and have a unique RDB_NAME. You must take the following steps to properly define the Application Server:

1. Select the LU name and RDB_NAME to be used by the DB2 for MVS/ESA Application Server. The process to record these names in DB2 for MVS/ESA and VTAM is the same process described in “Defining the Local System” on page 6. The RDB_NAME you choose for DB2 for MVS/ESA must be supplied to all end users and Application Requesters that require connectivity to the Application Server.
2. Register the NETID.LUNAME value for the DB2 for MVS/ESA Application Server with each Application Requester requiring access, so the Application Requester

can route SNA requests to the DB2 for MVS/ESA server. This is true even in cases where the Application Requester is able to perform dynamic network routing, because the Application Requester must know the NETID.LUNAME before dynamic network routing can be used.

3. Provide the DRDA default TPN (X'07F6C4C2') to each Application Requester because DB2 for MVS/ESA automatically uses this value.
4. Create an entry in the VTAM mode table for each mode name that is requested by an Application Requester. These entries describe the RU sizes, pacing window size, and class of service for each mode name.
5. Define session limits for the Application Requesters that connect with the DB2 for MVS/ESA Application Server. The VTAM APPL statement defines default session limits for all partner systems. If you want to establish unique defaults for a particular partner, you can use the SYSIBM.SYSLUMODES table of the communications database (CDB).

   See “Setting RU Sizes and Pacing” on page 15 about how to review your VTAM network.
6. Create entries in the DB2 for MVS/ESA CDB to identify which Application Requesters are allowed to connect to the DB2 for MVS/ESA Application Server. Two basic approaches to define the CDB entries for the Application Requesters in the network are:

   a. You can insert a row in SYSIBM.SYSLUNAMES that provides default values to use for any LU not specifically described in the CDB (the default row contains blanks in the LUNAME column). This approach allows you to define specific attributes for some of the LUs in your network, while establishing defaults for all other LUs.

      For example, you can allow the DALLAS system (another DB2 for MVS/ESA system) to send already-verified distributed database requests (LU 6.2 SECURITY=SAME), while requiring database manager systems to send passwords. Furthermore, you might not want to record an entry in the CDB for each database manager system, especially if there is a large number of these systems. Figure 10 shows how the CDB can be used to specify SECURITY=SAME for the DALLAS system, while enforcing SECURITY=PGM for all other requesters.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', ' ');
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES (' ', ' ', 'C', 'N', 'N', ' ');
```

Figure 10. Establishing Defaults for Application Requester Connections

   b. You can use the CDB to individually authorize each Application Requester in the network, by setting the CDB in one of these ways:

- Do not record a default row in SYSIBM.SYSLUNAMES. When the default row (the row containing a blank LU name) is not present, DB2 for MVS/ESA requires a row in SYSIBM.SYSLUNAMES containing the LU name for each application requester that attempts to connect. If the matching row is not found in the CDB, the Application Requester is denied access.
- Record a default row in SYSIBM.SYSLUNAMES that specifies come-from checking is required (USERNAMES column set to 'I' or 'B'). This causes DB2 for MVS/ESA to limit access to Application Requesters and end users identified in the SYSIBM.SYSUSERNAMES table, as described in “Come-From Checking” on page 28 . You might want to use this approach if your name translation rules require a row with a blank LU name in SYSIBM.SYSLUNAMES, but you do not want DB2 for MVS/ESA to use this row to allow unrestricted access to the DB2 for MVS/ESA Application Server.

In Figure 11, no row contains blanks in the LUNAME column, so DB2 for MVS/ESA denies access to any LU other than LUDALLAS or LUNYC.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', ' ');
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', ' ');
```

Figure 11. Identifying Individual Application Requester Connections

Defining Secondary Servers

DB2 for MVS/ESA does not implement a database server as defined in DRDA. Instead, DB2 for MVS/ESA provides secondary servers that provide access to multiple DB2 for MVS/ESA systems in a single unit of work using system-directed access.

SQL differences: The SQL supported by system-directed access differs significantly from DRDA remote unit of work:

- The SQL CONNECT statement is not used to establish a connection to a secondary server. Instead, the server is accessed by specifying three-part SQL object names. For example, the following SQL statement is routed to the CHICAGO system-directed access server:

  ```
  SELECT * FROM CHICAGO.USER.TABLE;
  ```

- SQL DDL statements (for example, CREATE) are not allowed.
- System-directed access does not support remote bind (for example, BIND PACKAGE), so you do not have to bind your application at the system-directed access server before attempting to execute the application.

- The SQL statements sent to a secondary server can be static or dynamic, but all the statements are issued dynamically. This occurs because the secondary server does not have a plan or package containing the application's SQL statements, so it is not possible for the server to choose database access paths in advance.
- A single SQL application can access multiple secondary servers simultaneously.
- More than one DB2 for MVS/ESA system can be the target of SQL updates for any given commit scope.
- An application can use multiple LU 6.2 conversations to a secondary server in a single commit scope. The DB2 for MVS/ESA Application Server usually creates one LU 6.2 conversation for each read-only SQL query. This allows the secondary server to anticipate the SQL application's FETCH requests, and send the answer set before it is actually requested by the application.

SQL object names: When the DB2 for MVS/ESA Application Server receives an SQL request, it examines the SQL object name to determine where the object resides in the network. DB2 for MVS/ESA accepts either one-, two-, or three-part SQL object names, where the name takes one of the following forms:

objectname specifies the name of a DB2 for MVS/ESA table, view, synonym, or alias.

authid.objectname specifies the owner of the object and the object name.

location.authid.objectname specifies the owning system, the owning user, and the name of the object.

If the location name (the first part of the three-part object name) matches the local DB2 for MVS/ESA system's RDB_NAME, the request identifies a local DB2 for MVS/ESA object.

If the location name does not match the local DB2 for MVS/ESA system's RDB_NAME, the DB2 for MVS/ESA Application Server reroutes the request to the system identified by the location name using system-directed access. The target system must be another DB2 for MVS/ESA system, because system-directed access is only supported between DB2 for MVS/ESA systems. System-directed access does not support any remote bind functions, so the application does not have to be bound at the server before executing the application. Figure 12 on page 26 summarizes the process used by DB2 for MVS/ESA to resolve SQL object names.

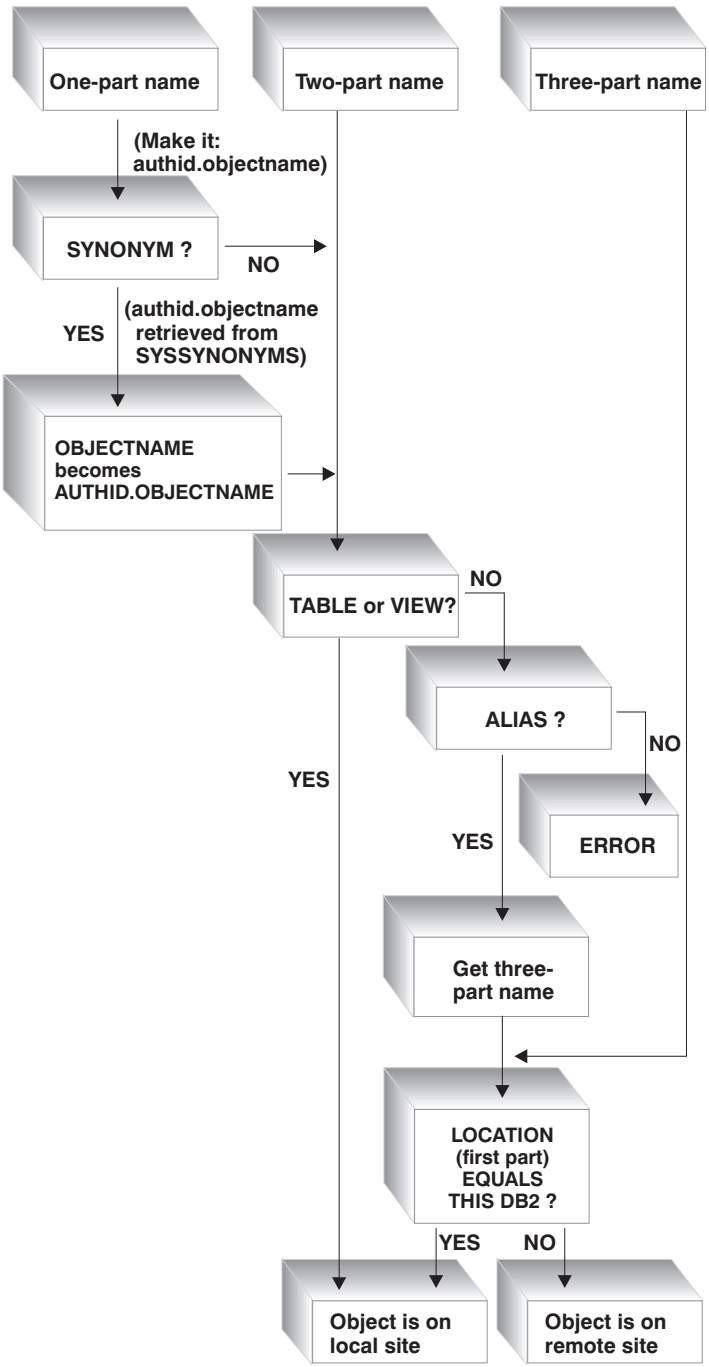

Figure 12. DB2 for MVS/ESA SQL Object Name Resolution

Server definition: If the DB2 for MVS/ESA Application Server is to reroute SQL requests, you must define each secondary server in the CDB and VTAM. Most of the definition process is similar to the process described in “Defining the Remote Systems” on page 11. To connect secondary servers, do the following:

1. Record the RDB_NAME and LU name values for each server in the CDB and VTAM. The TPN value used by system-directed access is different than the DRDA default value. However, this difference is not important because DB2 for MVS/ESA automatically chooses the correct value.
2. Define the security requirements in SYSIBM.SYSLUNAMES for each secondary server. This process is described in “Provide Security” on page 16.
3. Define the mode name (or names) used between the DB2 for MVS/ESA Application Server and the secondary servers, and place these mode names in the VTAM mode table. The default mode name is IBMDB2LM.
4. Define the session limits for each secondary server. The process used to establish the session limits is the same as the process described in “Defining the Local System” on page 6. However, system-directed access can establish multiple conversations for each SQL application. You might need to establish higher session limits for the system-directed access connections than you establish for DRDA connections. See “Connecting Distributed Database Systems” in the *DB2 Administration Guide* for specific details on how to calculate the number of LU 6.2 sessions required by system-directed access applications.

As the owner of the database resources, the secondary server controls database security for SQL objects residing at the server. However, this responsibility is shared with the DB2 for MVS/ESA Application Server making the request. The server controls access to SQL objects as follows:

- The secondary server does not have a copy of the DB2 for MVS/ESA plan, so it depends on the requesting DB2 for MVS/ESA Application Server to verify that the end user is allowed to execute the package at the requesting system (the Application Server).
- Static SQL statements are executed dynamically at the secondary server using the privileges granted to the person owning the DB2 for MVS/ESA package at the requesting DB2 for MVS/ESA Application Server.
- Dynamic SQL statements are executed using the privileges granted to the end user at the Application Requester.

Provide Security

When an Application Requester routes a distributed database request to the DB2 for MVS/ESA Application Server, the following security considerations can be involved:

- Come-from checking
- Selection of end user names
- Network security parameters
- Database manager security

- Security enforced by an external security subsystem

Come-From Checking

When the DB2 for MVS/ESA Application Server receives an end user name from the Application Requester, the Application Server can restrict the end user names received from a given Application Requester. This is accomplished through the use of *come-from* checking. Come-from checking allows the Application Server to specify that a given user ID is only allowed to be used by particular partners. For example, the Application Server can restrict JONES to “come from” DALLAS. If another Application Requester (other than DALLAS) attempts to send the name JONES to the Application Server, the Application Server can reject the request because the name did not come from the correct network location.

DB2 for MVS/ESA implements come-from checking as part of inbound end user name translation, which is described in the next section.

Selecting End User Names

The user ID passed by the Application Requester might not be unique throughout the entire SNA network. The DB2 for MVS/ESA Application Server might need to perform inbound name translation to create unique end user names throughout the SNA network. Similarly, the DB2 for MVS/ESA Application Server might need to perform outbound name translation to provide a unique end user name to the secondary servers involved in the application (see “Provide Security” on page 16 for information concerning outbound end user name translation).

Inbound name translation is enabled by setting the USERNAMES column of the SYSIBM.SYSLUNAMES table to `'I'` (inbound translation) or `'B'` (both inbound and outbound translation). When inbound name translation is in effect, DB2 for MVS/ESA translates the user ID sent by the Application Requester and the DB2 for MVS/ESA plan owner's name (if the Application Requester is another DB2 for MVS/ESA system).

If the Application Requester sends both a user ID and a password on the APPC ALLOCATE verb, the user ID and password are validated before the user ID is translated. The PASSWORD column in SYSIBM.SYSUSERNAMES is not used for password validation. Instead, the user ID and password are presented to the external security system (RACF or a RACF-equivalent product) for validation.

When the incoming user ID on the ALLOCATE verb is verified, DB2 for MVS/ESA has authorization exits you can use to provide a list of secondary AUTHIDs and perform additional security checks. See the *DB2 Administration Guide*, for details.

The inbound name translation process searches for a row in the SYSIBM.SYSUSERNAMES table, which must fit one of the patterns shown in the following precedence list (TYPE.AUTHID.LUNAME):

1. I.AUTHID.LUNAME—A specific end user from a specific Application Requester
2. I.AUTHID.blank—A specific end user from any Application Requester
3. I.blank.LUNAME—Any end user from a specific Application Requester

If no row is found, remote access is denied. If a row is found, remote access is allowed and the end user's name is changed to the value provided in the NEWAUTHID column, with a blank NEWAUTHID value indicating that the name is unchanged. Any DB2 for MVS/ESA resource authorization checks (for example, SQL table privileges) made by DB2 for MVS/ESA are performed on the translated end user names, rather than on the original user names.

When the DB2 for MVS/ESA Application Server receives an end user name from the Application Requester, several objectives can be accomplished by using the DB2 for MVS/ESA inbound name translation capability:

- You can change an end user's name to make it unique. For example, the following SQL statements translate the end user name JONES from the NEWYORK application requester (LUNAME LUNYC) to a different name (NYJONES).

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'JONES', 'LUNYC', 'NYJONES', ' ');
```

- You can change the end user's name so that a group of end users are all represented by a single name. For example, you might want to represent all users from the NEWYORK Application Requester (LUNAME LUNYC) with the user name NYUSER. This allows you to grant SQL privileges to the name NYUSER and to control the SQL access given to users from NEWYORK.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUNYC', 'NYUSER', ' ');
```

- You can restrict the end user names transmitted by a particular Application Requester. This use of end user name translation accomplishes the come-from check described in “Come-From Checking” on page 28. For example, the SQL statements that follow allow only SMITH and JONES as end user names from the NEWYORK Application Requester. Any other name is denied access, because it is not listed in the SYSIBM.SYSUSERNAMES table.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'SMITH', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'JONES', 'LUNYC', ' ', ' ');
```

- You can restrict the Application Requesters allowed to connect to the DB2 for MVS/ESA Application Server. This is yet another feature of come-from checking. The following example accepts any end user name sent by the NEWYORK Application Requester (LUNYC) or the CHICAGO Application Requester (LUCHI). Other Application Requesters are denied access, because the default SYSIBM.SYSLUNAMES row specifies inbound name translation for all inbound requests.

```
INSERT INTO SYSIBM.SYSLUNAMES
     (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES (' ', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.SYSUSERNAMES
     (TYPE, AUTHID, LUNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUCHI', ' ', ' ');
```

Provide Network Security

LU 6.2 provides three major network security features:

- Session-level security
- Conversation-level security
- Encryption

"Network Security" on page 18 discusses how to specify session-level security and encryption with DB2 for MVS/ESA. The DB2 for MVS/ESA Application Server uses session-level security and encryption in exactly the same manner as the DB2 for MVS/ESA Application Requester.

The only remaining network security consideration is SNA conversation-level security. Some aspects of conversation-level security are unique for a DB2 for MVS/ESA Application Server. The DB2 for MVS/ESA Application Server plays two distinct roles in network security:

- As a requester to secondary servers, the DB2 for MVS/ESA Application Server is responsible for issuing APPC requests that contain the SNA conversation-level security parameters required by the secondary servers. The DB2 for MVS/ESA Application Server uses the USERNAMES column of the SYSIBM.SYSLUNAMES table and the SYSIBM.SYSUSERNAMES table to define the SNA conversation level security requirements for each secondary server. The details of these definitions are identical to those in "Network Security" on page 18.
- As the server for the Application Requester, the DB2 for MVS/ESA Application Server dictates the SNA conversation level security requirements for the Application Requester. DB2 for MVS/ESA uses the USERSECURITY column of the SYSIBM.SYSLUNAMES table to determine the conversation security required from each Application Requester in the network. The following values are used in the USERSECURITY column:

C

This indicates that DB2 for MVS/ESA requires the Application Requester to send a user ID and password (LU 6.2 SECURITY=PGM) with each distributed database request. If the ENCRYPTPSWDS column in SYSIBM.SYSLUNAMES contains 'Y', DB2 for MVS/ESA assumes the password is already in RACF encrypted format (this is only possible for a DB2 for MVS/ESA Application Requester). If the ENCRYPTPSWDS column does not contain 'Y', DB2 for MVS/ESA expects the password in the standard LU 6.2 format (EBCDIC character representation). In either case, DB2 for MVS/ESA passes the user ID and password values to the security subsystem for validation. You must have a security subsystem that provides APPC user ID and password verification; for example, RACF has the capability to verify APPC user IDs and passwords. If the security subsystem rejects the user ID-password pair, distributed database access is denied.

Any other value

This indicates the Application Requester is allowed to send either an already-verified user ID (LU 6.2 SECURITY=SAME) or a user ID and password (LU 6.2 SECURITY=PGM). If a user ID and password are sent, DB2 for MVS/ESA processes them as described for 'C' above. If the request contains only a user ID, the security subsystem is called to authenticate the user unless the SYSUSERNAMES table is used to manage inbound user IDs.

If a security violation is discovered, LU 6.2 requires the DB2 for MVS/ESA Application Server to return the SNA security failure sense code ('080F6051'X) to the Application Requester. Because this sense code does not describe the cause of the failure, DB2 for MVS/ESA provides two methods for recording the cause of distributed security violations:

- A DSNL030I message is produced, which provides the requester's LUWID and a DB2 reason code describing the failure. DSNL030I also includes the AUTHID, if known, that was sent from the application request that was rejected.
- An alert is recorded in the NETVIEW hardware monitor database, which contains the same information provided in the DSNL030I message.

Database Manager Security

As the owner of database resources, the DB2 for MVS/ESA Application Server controls the database security functions for SQL objects residing at the DB2 for MVS/ESA Application Server. Access to DB2 for MVS/ESA-managed objects is controlled by privileges, which are granted to users by the DB2 for MVS/ESA administrator or the owners of individual objects. The two basic classes of objects that the DB2 for MVS/ESA Application Server controls are:

- **Packages**— Individual end users are authorized to create, replace, and run packages with the DB2 for MVS/ESA GRANT statement. When an end user owns a package, that user can automatically run or replace the package. Other end users must be specifically authorized to run a package at the DB2 for MVS/ESA Application Server with the GRANT statement. USE can be granted to individual end users or to PUBLIC, which allows all end users to run the package.

When an application is bound to DB2 for MVS/ESA, the package contains the SQL statements contained in the application program. These SQL statements are classified as:

Static SQL

Static SQL means that the SQL statement and the SQL objects referenced by the statement are known at the time the application is bound to DB2 for MVS/ESA. The person creating the package must have authority to execute each of the static SQL statements contained in the package.

When end users are granted authority to execute a package, they automatically have authority to execute each of the static SQL statements contained in the package. Thus, end users do not need any DB2 for MVS/ESA table privileges if the package they execute contains only static SQL statements.

Dynamic SQL

Dynamic SQL describes an SQL statement that is not known until the program executes. In other words, the SQL statement is built by the program and dynamically bound to DB2 for MVS/ESA with the SQL PREPARE statement. When an end user executes a dynamic SQL statement, the user must have the table privileges required to execute the SQL statement. Because the SQL statement is not known at the time the plan or package is created, the end user is not automatically given the required authority by the package owner.

- **SQL objects**— These are tables, views, synonyms, or aliases. DB2 for MVS/ESA users can be granted various levels of authority to create, delete, change, or read individual SQL objects. This authority is required to bind static SQL statements or to execute dynamic SQL statements.

When you create a package, the DISABLE/ENABLE option allows you to control which DB2 for MVS/ESA connection types can run the package. You can use RACF and DB2 for MVS/ESA security exit routines to selectively allow end users to use DDF. You can use RLF to specify limits on processor time for remote binds and dynamic SQL executions.

Consider a DB2 for MVS/ESA package named MYPKG, which is owned by JOE. JOE can allow SAL to execute the package by issuing the DB2 for MVS/ESA GRANT USE statement. When SAL executes the package, the following occurs:

- DB2 for MVS/ESA verifies that SAL was given USE authority for the package.
- SAL can issue every static SQL statement in the package because JOE had the required SQL object privileges to create the package.
- If the package has dynamic SQL statements, SAL must have SQL table privileges of her own. For example, SAL cannot issue SELECT * FROM JOE.TABLE5 unless she is granted read access to JOE.TABLE5.

Security Subsystem

The DB2 for MVS/ESA Application Server use of the security subsystem (RACF or a RACF-equivalent product) is dependent on how you define the inbound name translation function in the SYSIBM.SYSLUNAMES table:

- If you specify 'I' or 'B' for the USERNAMES column, inbound name translation is active, and DB2 for MVS/ESA assumes that the DB2 for MVS/ESA administrator is using inbound name translation to perform part of the system security enforcement. The external security subsystem is called only if the Application Requester sends a request containing both user ID and password (SECURITY=PGM). You must have a security subsystem that provides APPC user ID and password verification; for example, RACF has the capability to verify APPC user IDs and passwords.

  If the request from the Application Requester contains only a user ID (SECURITY=SAME), the external security system is not called at all, because the inbound name translation rules define which users are allowed to connect to the DB2 for MVS/ESA Application Server.

- If you specify something other than 'I' or 'B' for the USERNAMES column, the following security subsystem checks are performed:
  - When a distributed database request is received from the Application Requester, DB2 for MVS/ESA calls the external security system to validate the end user's user ID (and password if it is provided).
  - The external security system is called to verify that the end user is authorized to connect to the DB2 for MVS/ESA subsystem.
- In either case, an authorization exit is driven to provide a list of secondary authorization IDs. For more information, see the *DB2 Administration Guide*.

Represent Data

You must ensure that your DB2 for MVS/ESA subsystem has the ability to convert from each application requester's CCSID to your DB2 for MVS/ESA subsystem's installation CCSID. Refer to “Represent Data” on page 21 for more information.

Chapter 2. Connecting DB2 for OS/390 in a DRDA Network

DB2 for OS/390 Version 5 is the IBM relational database management system for OS/390 systems. This chapter does not address earlier releases. See Chapter 1, “Connecting DB2 for MVS/ESA in a DRDA Network” on page 1.

This chapter describes how to connect DRDA Application Requesters (such as DB2 Connect Version 5) to a DB2 for OS/390 Version 5 Application Server, and how to set up DB2 for OS/390 Application Requesters to communicate with DRDA Application Servers such as DB2 Universal Database Version 5 on other systems. Throughout this chapter “DB2 for OS/390” refers to DB2 for OS/390 Version 5.

The primary emphasis of the information in this chapter is on connecting unalike DRDA systems to DB2 for OS/390 using SNA network connections. However, DB2 for OS/390 Version 5 has also introduced support for database communications using native TCP/IP connections (not using AnyNet), and some discussion of using TCP/IP connections is also included. For more detailed information about using setting up and using TCP/IP connections, please refer to *DB2 for OS/390 Version 5 Installation Guide*, and *DRDA Support for TCPIP with DB2 for OS/390 Version 5 and DB2 Universal Database Version 5*.

For more information about connecting two DB2 for OS/390 systems, or more detailed information describing how to define DRDA connections to DB2 for OS/390, see the discussion of connecting distributed database systems in the *DB2 for OS/390 Administration Guide*.

Notes:

1. With the AnyNet Feature of VTAM Version 4 Release 2, you can run APPC over a TCP/IP network. The AnyNet Feature consists of AnyNet/MVS, which runs in a host, and AnyNet/2, which runs in a workstation and is downloaded from the host. Any APPC application is accessible to end users in a TCP/IP network without change to the application. Using APPC over TCP/IP, an application program on OS/390 can communicate with another APPC application program running with AnyNet APPC over TCP/IP on another system. See *VTAM AnyNet Feature for V4R2 Guide to SNA over TCP/IP* for more information.
2. This chapter does not contain any information about using DCE.

DB2 for OS/390

Figure 13 on page 36 shows an OS/390 system running a single copy of DB2 for OS/390. It is also possible to run multiple copies of DB2 for OS/390 on a single system. To identify copies of DB2 for OS/390 within a given system (or copies of DB2 for OS/390 within a JES complex), each DB2 system is given a *subsystem name*, a one- to four- character string unique within the JES complex. In Figure 13 on page 36, the DB2 for OS/390 subsystem name is *xxxx*. Three of the OS/390 address space names are prefixed by the DB2 for OS/390 subsystem name. These three address spaces make up the DB2 for OS/390 product.

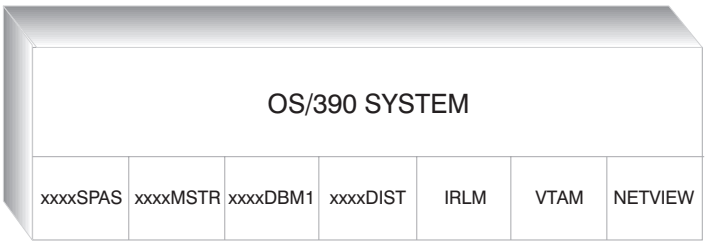

Figure 13. OS/390 Address Spaces used by DB2 for OS/390

Figure 13 shows the OS/390 address spaces involved in distributed database processing with DB2 for OS/390. These address spaces work together to allow DB2 for OS/390 users to access local relational databases and communicate with remote DRDA systems. The purpose of each address space is as follows:

xxxxSPAS The DB2 stored procedures address space.

xxxxMSTR The system services address space for the DB2 for OS/390 product responsible for starting and stopping DB2 for OS/390, and controlling local access to DB2 for OS/390.

xxxxDBM1 The database services address space responsible for accessing relational databases controlled by DB2 for OS/390. This is where the input and output to database resources is performed on behalf of SQL application programs.

xxxxDIST The portion of DB2 for OS/390 that provides distributed database capabilities; also known as the *Distributed Data Facility* (DDF). When a distributed database request is received, DDF passes the request to *xxxx*DBM1, so that the required database I/O operations can be performed.

IRLM The lock manager used by DB2 for OS/390 to control access to database resources.

VTAM IBM Communications Server for OS/390 SNA functions (VTAM). DDF can use SNA or TCP/IP to perform distributed database communications on behalf of DB2 for OS/390. No address space is shown for TCP/IP in this diagram.

NETVIEW The network management focal point product on OS/390 systems. When errors occur during distributed database processing, DDF records error information (also known as *alerts*) in the NetView hardware monitor database. System administrators can use NetView to examine the errors stored in the hardware monitor database, or provide automated command procedures to be invoked when alert conditions are recorded.

NetView can also be used to diagnose VTAM communication errors. For more information, see the *Distributed Relational Database Architecture Problem Determination Guide*.

Figure 13 on page 36 does not show any SQL application programs. When an application program uses DB2 to issue SQL statements, the application program must attach to the DB2 for OS/390 product in one of the following ways:

TSO Batch jobs and end users logged on to TSO are connected to DB2 for OS/390 through the TSO attach facility. This is the technique used to connect SPUFI and most QMF applications to DB2 for OS/390.

CICS/ESA When a CICS/ESA application issues SQL calls, the CICS/ESA product uses the CICS attach interface to route SQL requests to DB2 for OS/390.

IMS/ESA Transactions running under the control of IMS/ESA use the IMS attach interface to pass SQL statements to DB2 for OS/390 for processing.

DDF The Distributed Data Facility is responsible for connecting distributed applications to DB2 for OS/390.

CAF The call attachment facility allows user-written subsystems to connect directly to DB2 for OS/390.

DB2 for OS/390 Implementation

DRDA defines types of distributed database management system functions. DB2 for OS/390 Version 5 supports remote unit of work. With remote unit of work, an application program executing in one system can access data at a remote DBMS using the SQL provided by that remote DBMS.

DB2 for OS/390 Version 5 also supports distributed unit of work. With distributed unit of work, an application program executing in one system can access data at multiple remote DBMSs using SQL provided by remote DBMSs. For more information on the types of distribution defined by DRDA, see *DRDA Connectivity Guide*.

As shown in Figure 14 on page 39, DB2 for OS/390 supports three configurations of distributed database connections using two access methods:

[1] *System-directed access* (also known as using *DB2 for OS/390 private protocol*) allows a DB2 for OS/390 requester to connect to one or more DB2 for OS/390 servers. The connection established between the DB2 for OS/390 requester and server does not adhere to the protocols defined in DRDA and cannot be used to connect non-DB2 for OS/390 products to DB2 for OS/390. This type of connection is established by coding three-part names or aliases in the application.

[2] *Application-directed access* allows a DB2 for OS/390 or non-DB2 for OS/390 requester to connect to one or more DB2 for OS/390 or non-DB2 for OS/390 application servers using DRDA protocols. The number of application servers that can be connected to the application requester at one time depends on the level of DB2 for OS/390 of the application requester. If the application requester is DB2 for

MVS/ESA V2R3, then only one application server can be connected at a time. This type of connection is established by coding SQL CONNECT statements in the application. If the application requester is DB2 for MVS/ESA V3R1, or DB2 for OS/390 V5R1 , then one or more application servers can be connected at a time.

[3] Application-directed and system-directed access can be used together to establish connections. You cannot connect using DRDA and system-directed storage in the same thread.

The term *secondary server* describes systems acting as servers to the application server.

If all systems in a configuration support two-phase commit, then distributed unit of work (multiple-site read and multiple-site update) is supported. If not all systems support two-phase commit, updates within a unit of work are either restricted to a single site that does not support two-phase commit, or to the subset of sites that support two-phase commit.

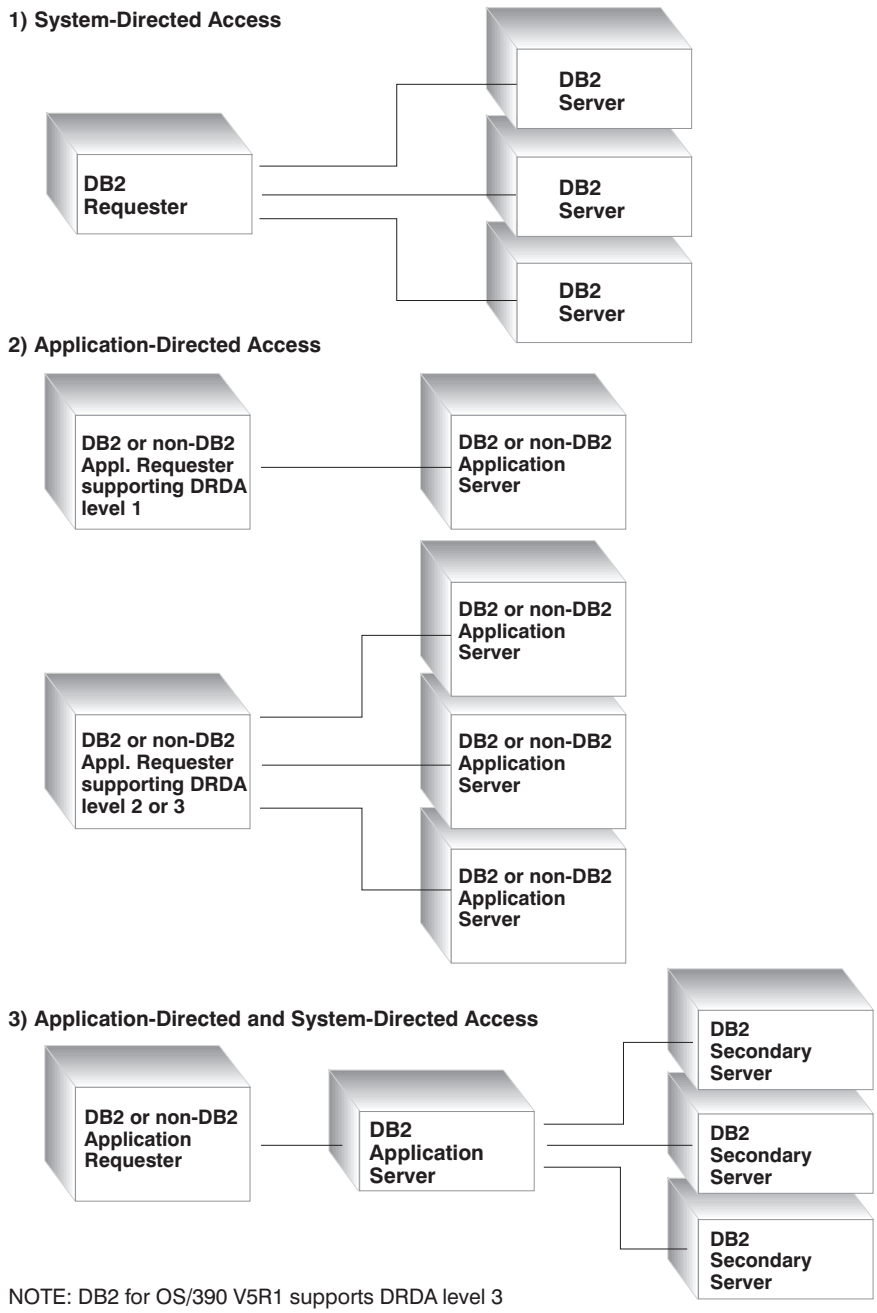

Figure 14. DB2 for OS/390 Distributed Connections

Table 2 on page 40 compares the DB2 for OS/390 distributed database connection types.

Table 2. Comparison of DB2 for OS/390 Distributed Database Connections

[1] System-Directed Access	**[2] Application-Directed Access (with all systems having two-phase commit)**	**[3] Application-Directed and System-Directed Accesses**
All partners must be DB2 for OS/390 systems	Can interconnect any two DRDA systems	Application requester can be any DRDA system; servers must be DB2 for OS/390 systems
Can connect directly to many partners	Can connect directly to many partners	Application requester connects directly to Application Servers; Application Servers can connect to many DB2 for OS/390 secondary servers
Each SQL application can have multiple conversations with each server	Each SQL application has one conversation with each server	SQL application has one conversation with each server; DB2 for OS/390 application server can establish many conversations to each server for the application
Can access both local and remote resources in one commit scope	Can access both local and remote resources in one commit scope	Application requester and Application Server can access local and remote data
More efficient at large queries and multiple concurrent queries	More efficient at SQL statements that are executed very few times in one commit scope	Application requester-Application Server connection behaves like **[2]**; secondary server connections behave like **[1]**
Can support static or dynamic SQL, but server dynamically binds static SQL the first time it is executed in a commit scope	Can issue static or dynamic SQL	Application requester and Application Server can issue static or dynamic SQL; secondary servers support static or dynamic SQL, but dynamically bind static SQL the first time it is executed in a commit scope
Limited to SQL INSERT, DELETE, and UPDATE statements, and to statements that support SELECT	Can use any statement supported by the system that executes the statement	Application servers supports any SQL; secondary servers support only DML SQL (for example, CREATE or ALTER)

Additional Security Enhancements

Extended Security Codes

Until DB2 for OS/390 Version 5.1, connect requests that provided user IDs or passwords could fail with `SQL30082` reason code 0, but no other indication as to what might be wrong.

DB2 for OS/390 Version 5.1 introduces an enhancement which provides support for extended security codes. Specifying extended security will provide additional diagnostics, such as (PASSWORD EXPIRED) in addition to the reason code.

In order to exploit this, the DB2 for OS/390 ZPARM installation parameter for extended security should be set to the value YES. Use the DB2 for OS/390 installation panel DSN6SYSP to set EXTSEC=YES. You can also use DDF panel 1 (DSNTIPR) to set this. The default value is EXTSEC=NO.

TCP/IP Security Already Verified

If you wish to provide support for the DB2 Universal Database security option AUTHENTICATION=CLIENT, then use DB2 for OS/390 installation panel DSNTIP4 (DDF panel 2) to set TCP/IP already verified security to YES.

ODBC Application Security

ODBC applications use dynamic SQL. This may create security concerns in some installations. DB2 for OS/390 introduces a new bind option DYNAMICRULES(BIND) that allows execution of dynamic SQL under the authorization of either the owner or the binder.

DB2 Universal Database Version 5.0 provides a new CLI/ODBC configuration parameter CURRENTPACKAGESET in the DB2CLI.INI configuration file. This should be set to a schema name that has the appropriate privileges. An SQL SET CURRENT PACKAGESET schema statement will automatically be issued after every connect for the application.

Use the ODBC Manager to update DB2CLI.INI. See *Installing and Configuring DB2 Clients* for further information.

Setting Up the Application Requester

DB2 for OS/390 implements the DRDA application requester support as an integral part of the DB2 for OS/390 Distributed Data Facility (DDF). DDF can be stopped independently from the local DB2 for OS/390 database management facilities, but it cannot run in the absence of the local DB2 for OS/390 database management support.

When DB2 for OS/390 acts as an Application Requester, it can connect to a DB2 for OS/390 Application Server or any other product that supports the DRDA architecture.

For the DB2 for OS/390 Application Requester to provide distributed database access, you need to do the following:

- "Provide Network Information" on page 42—The Application Requester must be able to accept RDB_NAME values and translate these values into SNA NETID.LUNAME or TCP/IP address values. DB2 for OS/390 uses the *DB2 for OS/390 Communications Database* (CDB) to register RDB_NAMEs and their corresponding network parameters. The CDB allows the DB2 for OS/390 Application Requester to pass the required information to the Communications Server when making distributed database requests over either SNA or TCP/IP connections.

- "Provide Security" on page 56— For remote database requests to be accepted by the Application Server, the Application Requester must provide the security information required by the server. DB2 for OS/390 uses the CDB and DCE, RACF, or other security subsystem to provide the required network security information.
- "Represent Data" on page 63—You must ensure that the CCSID of the application requester is compatible with the application server.

Provide Network Information

Much of the processing in a distributed database environment requires exchanging messages with other locations in your network. For this processing to be performed correctly, you need to do the following:

1. Define the local system
2. Define the remote systems
3. Define the communications (for either SNA or TCP/IP connections)
4. Set RU sizes and pacing (for SNA connections only)

See either "Defining the Local System (SNA)," or "Defining the Local System (TCP/IP)" on page 47 .

Defining the Local System (SNA)

Each program in the SNA network is assigned a NETID and an LU name, so your DB2 for OS/390 Application Requester must have a NETID.LUNAME value (assigned through VTAM) when it connects to the network. Because the DB2 for OS/390 Application Requester is integrated into the local DB2 for OS/390 database management system, the Application Requester must also have an RDB_NAME. In the DB2 for OS/390 publications, DB2 for OS/390 refers to the RDB_NAME as a *location* name.

Define the DB2 for OS/390 Application Requester to the SNA network as follows:

1. Select an LU name for your DB2 for OS/390 system. The NETID for your DB2 for OS/390 system is automatically obtained from VTAM when DDF starts.
2. Define the LU name and location name in the DB2 for OS/390 *bootstrap data set* (BSDS). (DB2 for OS/390 restricts the location name to 16 characters.)
3. Create a VTAM APPL definition to register the selected LU name with VTAM.
4. Ensure that Extended Security is set to YES. See "Additional Security Enhancements" on page 40.

Configuring the DDF BSDS: DB2 for OS/390 reads the BSDS during startup processing to obtain system installation parameters. One of the records stored in the BSDS is called the *DDF record*, because it contains the information used by DDF to connect to VTAM. This information consists of the following:

- The location name for the DB2 for OS/390 system
- The LU name for the DB2 for OS/390 system
- The password used when connecting the DB2 for OS/390 system to VTAM

You can supply the DDF BSDS information to DB2 for OS/390 in two ways:

- Use the DDF installation panel DSNTIPR when you first install DB2 for OS/390 to provide the required DDF BSDS information. Many of the install parameters are not discussed here because it is more important to know how to connect DB2 for OS/390 to VTAM. Figure 15 shows how to use the installation panel to record location name SYDNEY, the LU name LUDBD1, and password PSWDBD1 in the DB2 for OS/390 BSDS.

```
                              DISTRIBUTED DATA FACILITY                      =
==> _

 Enter data below:

  1 DDF STARTUP OPTION   ===> AUTO      NO, AUTO, or COMMAND
  2 DB2 LOCATION NAME    ===> NEW_YORK3         The name other DB2s use to
                                                refer to this DB2
  3 DB2 NETWORK LUNAME   ===> NYM2DB2   The name VTAM uses to refer to this DB2
  4 DB2 NETWORK PASSWORD ===> PSWDBD1   Password for DB2's VTAM application
  5 RLST ACCESS ERROR    ===> NOLIMIT   NOLIMIT, NORUN, or 1-5000000
  6 RESYNC INTERVAL      ===> 3         Minutes between resynchronization period
  7 DDF THREADS          ===> ACTIVE    (ACTIVE or INACTIVE) Status of a
                                        database access thread that commits or
                                        rolls back and holds no database locks
                                        or cursors
  8 DB2 GENERIC LUNAME   ===>           Generic VTAM LU name for this DB2
                                        subsystem or data sharing group
  9 IDLE THREAD TIMEOUT  ===> 120       0 or seconds until dormant server ACTIVE
                                        thread will be terminated (0-9999)
 10 EXTENDED SECURITY    ===> YES       Allow change password and descriptive
                                        security error codes.  YES or NO.
 PRESS:  ENTER to continue   RETURN to exit   HELP for more information
```

Figure 15. DB2 for OS/390 Installation Panel DSNTIPR

- If DB2 for OS/390 is already installed, you can use the change log inventory utility (DSNJU003) to update the information in the BSDS.

  Figure 16 on page 44 shows how to update the BSDS with location name NEW_YORK3, the LU name NYM2DB2, and password PSWDBD1.

```
//SYSADMB JOB ,'DB2 5.1 JOB',CLASS=A
//*
//*        CHANGE LOG INVENTORY:
//*        UPDATE BSDS WITH
//*             - DB2 LOCATION NAME FOR NEW_YORK3
//*             - VTAM LUNAME (NYM2DB2)
//*             - DB2/VTAM PASSWORD
//*
//DSNBSDS EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DSN510.DSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DSNC510.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DSNC510.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 DDF    LOCATION=NEW_YORK3,LUNAME=NYM2DB2,PASSWORD=PSWDBD1
//*
```

Figure 16. Sample Bootstrap Data Set DDF Definition (for VTAM)

When DDF is started (either automatically at DB2 for OS/390 startup or by the DB2 for OS/390 START DDF command), it connects to VTAM, passing the LU name and password to VTAM. VTAM recognizes the DB2 for OS/390 system by checking the LU name and password (if a VTAM password is required) with the values defined in the DB2 for OS/390 VTAM APPL statement. The VTAM password is used to verify that DB2 for OS/390 is authorized to use the specified LU name on the VTAM system. The VTAM password is not transmitted through the network, and it is not used to connect other systems in the network to DB2 for OS/390.

If VTAM does not require a password, omit the PASSWORD= keyword on the change log inventory utility. The absence of the keyword indicates that no VTAM password is required.

Creating a VTAM APPL definition: After you define the VTAM LU name and password to DB2 for OS/390, you need to register these values with VTAM. VTAM uses the APPL statement to define local LU names. Figure 17 on page 45 shows how to define the LU name NYM2DB2 to VTAM.

```
 DB2APPLS VBUILD TYPE=APPL
 *
 *--------------------------------------------------------------------*
 *                                                                    *
 *           APPL DEFINITION FOR THE NEW_YORK3 DB2 SYSTEM             *
 *                                                                    *
 *--------------------------------------------------------------------*
 *
NYM2DB2   APPL  APPC=YES,                                            X
                AUTH=(ACQ),                                          X
                AUTOSES=1,                                           X
                DMINWNL=10,                                          X
                DMINWNR=10,                                          X
                DSESLIM=20,                                          X
                EAS=9999,                                            X
                MODETAB=RDBMODES,                                    X
                PRTCT=PSWDBD1,                                       X
                SECACPT=ALREADYV,                                    X
                SRBEXIT=YES,                                         X
                VERIFY=NONE,                                         X
                VPACING=2,                                           X
                SYNCLVL=SYNCPT,                                      X
                ATNLOSS=ALL                                          X
```

Figure 17. Sample VTAM APPL Definition for DB2 for OS/390

Many keywords are available on the VTAM APPL statement. The meaning of the keywords is discussed in detail in the *DB2 for OS/390 Administration Guide*. The only keywords discussed here address topics in this book. The keywords of interest in Figure 17 are described as follows:

NYM2DB2

VTAM uses the APPL statement label as the LU name. In this case, the LU name is NYM2DB2. The APPL syntax does not allow room for a complete NETID.LUNAME value. The NETID value is not specified on the VTAM APPL statement, because all VTAM applications are automatically assigned the NETID for the VTAM system.

AUTOSES=1

The number of SNA contention winner sessions that start automatically when an APPC Change Number of Sessions (CNOS) request is issued. A nonzero value must be supplied with AUTOSES to inform DB2 for OS/390 in all cases when VTAM CNOS processing fails.

You do not have to automatically start all the APPC sessions between any two distributed database partners. If the AUTOSES value is less than the contention winner limit (DMINWNL), VTAM delays starting the remaining SNA sessions until they are required by a distributed database application.

DMINWNL=10

The number of sessions on which this DB2 for OS/390 system is the contention winner. The DMINWNL parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.LUMODES table in the DB2 for OS/390 CDB.

DMINWNR=10

The number of sessions on which the partner system is the contention winner. The DMINWNR parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.LUMODES table in the DB2 for OS/390 CDB.

DSESLIM=20

The total number of sessions (winner and loser sessions) you can establish between DB2 for OS/390 and another distributed system for a specific mode group name. The DSESLIM parameter is the default for CNOS processing, but can be overridden for any given partner by adding a row to the SYSIBM.LUMODES table in the DB2 for OS/390 CDB.

If the partner cannot support the number of sessions requested on the DSESLIM, DMINWNL, or DMINWNR parameters, the CNOS process negotiates new values for these parameters that are acceptable to the partner.

EAS=9999

An estimate of the total number of sessions that this VTAM LU requires.

MODETAB=RDBMODES

Identifies the VTAM MODE table where each DB2 for OS/390 mode name exists.

PRTCT=PSWDBD1

Identifies the VTAM password to use when DB2 for OS/390 attempts to connect to VTAM. If the PRTCT keyword is omitted, no password is required, and you should omit the PASSWORD= keyword from the DB2 for OS/390 change log inventory utility.

SECACPT=ALREADYV

Identifies the highest SNA conversation-level security value accepted by this DB2 for OS/390 system when it receives a distributed database request from a remote system. The ALREADYV keyword indicates this DB2 for OS/390 system can accept three SNA session security options from other DRDA systems that request data from this DB2 for OS/390 system:

- SECURITY=SAME (an already-verified request that contains only the requester's user ID).
- SECURITY=PGM (a request containing the requester's password, or a PassTicket).
- SECURITY=NONE (a request containing no security information). DB2 for OS/390 rejects DRDA requests that specify SECURITY=NONE.

It is best to always specify SECACPT=ALREADYV, because the SNA conversation security level for each DB2 for OS/390 partner is taken from the DB2 for OS/390 CDB (the USERSECURITY column of the SYSIBM.LUNAMES table).

SECACPT=ALREADYV gives you the most flexibility in selecting values for USERSECURITY.

VERIFY=NONE
Identifies the level of SNA session security (partner LU verification) required by this DB2 for OS/390 system. The NONE value indicates that partner LU verification is not required.

DB2 for OS/390 does not restrict your choice for the VERIFY keyword. In a non-trusted network, VERIFY=REQUIRED is recommended. VERIFY=REQUIRED causes VTAM to reject partners that cannot perform partner LU verification. If you choose VERIFY=OPTIONAL, VTAM performs partner LU verification only for those partners that provide the support.

VPACING=2
Sets the VTAM pacing count to 2.

SYNCLVL=SYNCPT
Indicates that DB2 for OS/390 is able to support two-phase commit. VTAM uses this information to inform the partner that two-phase commit is available. If this keyword is present, DB2 for OS/390 automatically uses two-phase commit if the partner can support it.

ATNLOSS=ALL
Indicates that DB2 for OS/390 needs to be informed each time a VTAM session ends. This ensures that DB2 for OS/390 performs SNA resynchronization when required.

DSESLIM, DMINWNL, and DMINWNR allow you to establish default VTAM session limits for all partners. For partners that have special session limit requirements, the SYSIBM.LUMODES table can be used to override the default session limits. For example, you might want to specify VTAM default session limits that are appropriate for your OS/2 systems. For other partners, you can create rows in the SYSIBM.LUMODES table to define the desired session limits. Consider these sample values:

```
DSESLIM=4,DMINWNL=0,DMINWNR=4
```

These parameters allow each partner to create up to four sessions with DB2 for OS/390, where the partner is the contention winner on each of the sessions. Because OS/2 creates the LU 6.2 conversations with DB2 for OS/390, by making OS/2 the contention winner on the sessions, you gain a small performance advantage. If OS/2 has an available contention winner session, it does not have to ask for permission to start a new LU 6.2 conversation.

Defining the Local System (TCP/IP)

For your convenience the information in this section has been summarized from *DB2 Connect Enterprise Edition Quick Beginnings*. For more detailed information, please see *DB2 for OS/390 Version 5 Installation Reference*, and *DRDA Support for TCP/IP with DB2 for OS/390 Version 5 and DB2 Universal Database Version 5*.

The steps required to define TCP/IP communications with DB2 for OS/390 are as follows:

1. TCP/IP communications must be enabled on DB2 for OS/390 and the partner system.
2. Two suitable TCP/IP port numbers must be assigned by your network administrator. As a default, DB2 for OS/390 uses port number 446 for database connections, and port number 5001 for resynchronization requests (two-phase commit).
3. The remote application server or application requester must use the same port numbers (or service names) as DB2 for OS/390.
4. Ensure that the TCP/IP already verified security option is set to `YES`. See "Additional Security Enhancements" on page 40.
5. The DB2 for OS/390 BSDS must include additional parameters. Figure 18 highlights the additional parameters required to enable TCP/IP communications.

```
//SYSADMB JOB ,'DB2 5.1 JOB',CLASS=A
//*
//*        CHANGE LOG INVENTORY:
//*        UPDATE BSDS WITH
//*             - DB2 LOCATION NAME FOR NEW_YORK3
//*             - VTAM LUNAME (NYM2DB2)
//*             - DB2/VTAM PASSWORD
//*
//*             - GENERIC LU NAME
//*             - TCP/IP PORT FOR DATABASE CONNECTIONS
//*             - TCP/IP PORT FOR RESYNCH OPERATIONS
//*
//DSNBSDS EXEC PGM=DSNJU003
//STEPLIB  DD  DISP=SHR,DSN=DSN510.DSNLOAD
//SYSUT1   DD  DISP=OLD,DSN=DSNC510.BSDS01
//SYSUT2   DD  DISP=OLD,DSN=DSNC510.BSDS02
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 DDF    LOCATION=NEW_YORK3,LUNAME=NTYM2DB2,PASSWORD=PSWDBD1,
        GENERICLU=name,PORT=446,RESPORT=5001
/*
//*
```

Figure 18. Sample Bootstrap Data Set DDF Definition (for TCP/IP)

Defining the Remote Systems

When a DB2 for OS/390 application requests data from a remote system, it searches the Communications Database (CDB) tables to find information about the remote system. The CDB is a group of SQL tables managed by the DB2 for OS/390 system administrator. As the DB2 for OS/390 system administrator, you can use SQL to insert rows in the CDB to describe each potential DRDA partner. A complete description of

the CDB and how it is used can be found in *DB2 for OS/390 Version 5 SQL Reference*, and *DB2 for OS/390 Version 5 Installation Guide*.

References to the CDB search for information including:

- The LU name and TPN (for SNA connections)
- TCP/IP address information (required for outbound TCP/IP SNA connections only)
- The network security information required by the remote site
- The session limits and mode names used to communicate with the remote site (for SNA connection)

Populating The Communications Database: No CDB updates are required if you will only use inbound TCP/IP database connections, so that if you plan to use DB2 for OS/390 only as a TCP/IP server, you do not need to populate the CDB, and default values can be used. However, if you will use inbound SNA connections, you must at least provide a single blank row in SYSIBM.LUNAMES. For example, to permit SNA database connection requests to be accepted from any incoming DB2 Connect LU, use an SQL command such as the following:

```
INSERT INTO SYSIBM.LUNAMES (LUNAME) VALUES ('        ')
```

When you will use DB2 for OS/390 as a requester the CDB must always be updated. You will need to insert rows in into the SYSIBM.LOCATIONS table, and either the SYSIBM.LUNAMES table (for SNA connections), or the SYSIBM.IPNAMES table (for TCP/IP connections).

Further, if you want to control inbound security requirements or inbound user-id translation for SNA connections, additional CDB updates may be required. Additional examples are provided as follows: “Provide Security” on page 56 describes how to define user security when setting up an application requester, and “Provide Security” on page 67 describes setting up an application server.

The *DB2 for OS/390 Administration Guide* discusses the requirements for updating CDB tables in more detail. After you populate the CDB, you can write queries that access data at remote systems. *DB2 for OS/390 Installation Reference* also provides further information about updating the CDB.

How the Communications Database Handles Requests: When sending a request, DB2 for OS/390 uses the LINKNAME column of the SYSIBM.LOCATIONS catalog table to determine which network protocol to use for the outbound database connection. To receive VTAM requests, you must select an LUNAME in DB2 for OS/390 installation panel DSNTIPR. To receive TCP/IP requests, you must select a DRDA port and a resynchronization port in DB2 for OS/390 installation panel DSNTIP5. TCP/IP uses the server's port number to pass network requests to the correct DB2 subsystem.

If the value in the LINKNAME column is found in the SYSIBM.IPNAMES table, TCP/IP is used for DRDA connections. If the value is found in SYSIBM.LUNAMES table, SNA is used. If the same name is in both SYSIBM.LUNAMES and SYSIBM.IPNAMES, TCP/IP is used to connect to the location.

Note: A requester cannot connect to a given location using both SNA and TCP/IP protocols. For example, if your SYSIBM.LOCATIONS specifies a LINKNAME of LU1, and if LU1 is defined in both the SYSIBM.IPNAMES and SYSIBM.LUNAMES table, TCP/IP is the only protocol used to connect to LU1 from this requester.

Communications Database Tables: The CDB consists of the following tables:

1. **SYSIBM.LOCATIONS**

   This table allows DB2 for OS/390 to determine the SNA or TCP/IP address information required when accessing each RDB_NAME selected by a DB2 for OS/390 application *for outbound requests*. The columns are:

   LOCATION The RDB_NAME of the remote system. DB2 for OS/390 limits the RDB_NAME value to 16 bytes, which is two bytes shorter than the 18-byte limit defined in DRDA.

   LINKNAME The LU name or TCP/IP attributes of the remote system.

   PORT TCP/IP port name or service name information (the default port name for DRDA is 446).

   TPN The APPC Transaction Program name (TPN) of the remote system. If the remote system is a DB2 for OS/390 system or the remote system uses the default DRDA TPN value (X'07F6C4C2'), an empty string can be used to specify the TPN because DB2 for OS/390 automatically chooses the correct value.

   If the remote system requires a TPN value other than the default TPN value, you must supply this value here.

2. **SYSIBM.LUNAMES**

   This table defines the network attributes of remote systems accessed using SNA connections. The columns are:

   LUNAME The LU name of the remote system.

   SYSMODENAME The VTAM logon mode name used to establish DB2 for OS/390-to-DB2 for OS/390 *intersystem* conversations for the DB2 for OS/390's *secondary server support* (system-directed access). A blank value in this column indicates IBMDB2LM should be used for DB2 for OS/390 system conversations.

   SECURITY_IN The network security acceptance options required of the remote system when this DB2 for OS/390 system acts as a server for the remote system (*inbound security* requirements). Values can be:

   - **V** signifying the option is "verify." An incoming connection request must include one of the following: a userid and password, a userid and RACF PassTicket, or a DCE security ticket.

- **A** signifying the option is "already verified." A request does not need a password, although a password is checked if it is sent. With this option, an incoming connection request is accepted if it includes any of the following: a userid, a userid and password, a userid and RACF PassTicket, or a DCE security ticket.

  If the USERNAMES column contains 'I' or 'B', RACF is not invoked to validate incoming connection requests that contain only a userid.

SECURITY_OUT The network security acceptance options required of the remote system when this DB2 for OS/390 system acts as a requester (*outbound security* requirements). Values can be:

- **A** signifying the option is "already verified." Outbound connection requests contain an authorization ID and no password. The authorization ID used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending upon the value of the USERNAMES column.
- **R** signifying the option is "RACF PassTicket." Outbound connection requests contain a userid and a RACF PassTicket. The server's LU name is used as the RACF PassTicket application name.

  The authorization ID used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending upon the value of the USERNAMES column.
- **P** signifying the option is "password." Outbound connection requests contain an authorization ID and a password. The password is obtained from the SYSIBM.USERNAMES table or RACF, depending upon the value specified in the ENCRYPTPWDS column.

  The USERNAMES column must specify 'B' or 'O'.

ENCRYPTPSWDS Whether passwords exchanged with this partner are encrypted. Encrypted passwords are only supported by DB2 for OS/390 requesters and servers.

MODESELECT Determines whether the SYSIBM.MODESELECT table is used to select a VTAM logon mode entry (mode name) based on the end user and application making the request. If this column contains a `'Y'`, the SYSIBM.MODESELECT table is used to obtain the mode name for each outbound distributed database request.

If MODESELECT contains anything other than a `'Y'`, the mode name IBMDB2LM is used for system-directed access requests, and the mode name IBMRDB is used for DRDA requests.

The MODESELECT column allows you to prioritize distributed database requests by specifying a VTAM class of service (COS) associated with the mode name.

USERNAMES The level of come-from checking and user ID translation required. This column also specifies the security parameters this DB2 for OS/390 subsystem uses when requesting data from the remote partner (*outbound security* requirements). USERNAMES can have the value I, O, or B (inbound-only, outbound-only, or both).

GENERIC Indicates whether DB2 for OS/390 should use its real or generic LU name.

3. **SYSIBM.LUMODES**

   This table is used to provide VTAM with LU 6.2 session limits (CNOS limits) for partner systems using APPC (SNA) connections. The columns are:

   LUNAME The LU name of the remote system.

   MODENAME The name of the VTAM logon mode whose limits are being specified. A blank value in the MODENAME column defaults to IBMDB2LM.

   CONVLIMIT The maximum number of active conversations between the local DB2 for OS/390 and the remote system for this logon mode. This value is used to override the DSESLIM parameter in the VTAM APPL definition statement for this logon mode, which supplies the default VTAM session limits for DB2 for OS/390.

   The value selected in CONVLIMIT is used during CNOS to set the DMINWNR and DMINWNL values to CONVLIMIT/2.

4. **SYSIBM.MODESELECT**

   This table allows you to specify different mode names for individual end users and DB2 for OS/390 applications. It is only used for SNA connections. Because each VTAM mode name can have an associated class of service (COS), you can use this table to assign network transmission priorities to distributed database applications based on a combination of AUTHID, PLANNAME, and LUNAME. The columns are:

   AUTHID The DB2 for OS/390 user's authorization ID (user ID). The default is blank, indicating the specified logon mode name applies to all authorization IDs.

   PLANNAME The plan name associated with the application requesting access to a remote database system. The default is blank, indicating that the specified logon mode name applies to all plan names. The plan name used for the BIND PACKAGE command is DSNBIND.

LUNAME The LU name associated with the remote database system.

MODENAME The name of the VTAM logon mode to use when routing a distributed database request to the indicated remote system. The default is blank, indicating that IBMDB2LM should be used for system-directed access conversations and IBMRDB should be used for DRDA conversations.

5. **SYSIBM.USERNAMES**

   This table is used to manage end user names by providing passwords, name translations, and come-from checking. DB2 for OS/390 refers to the end user's name as an authorization ID. Most other products refer to this name as a user ID.

   With this table, you can use name translation to force different values to be used for user IDs connections and the DB2 for OS/390 authorization ID. The name translation process is allowed for requests to a remote system (*outbound* requests) and for requests coming from a remote system (*inbound* requests). If passwords are not encrypted, this table is the source of the end user's password when both user ID and password are sent to a remote site. The columns are:

   TYPE The description of how the row is to be used (whether it is a row describing name translations for outbound or inbound/come-from checking requests).

   I signifies inbound connections, **O** signifies outbound connections.

   Use "O" for TCP/IP connections (inbound ID translation and come-from checking are not done for TCP/IP requests).

   AUTHID For outbound name translation, this is the DB2 for OS/390 authorization ID to translate. For inbound name translation, this is the SNA user ID to translate. In either case, a blank AUTHID value applies to all authorization IDs or user IDs.

   LINKNAME Identifies the VTAM or TCP/IP network locations associated with this row. A blank value in this column indicates this name translation rule applies to any TCP/IP or SNA partner.

   If a nonblank LINKNAME is specified, one or both of the following statements must be true:

   - A row exists in SYSIBM.LUNAMES whose LUNAME matches the value specified in the SYSIBM.USERNAMES LINKNAME column. This row specifies the VTAM site associated with this name translation rule.
   - A row exists in SYSIBM.IPNAMES whose LINKNAME matches the LINKNAME matches the value specified in the SYSIBM.USERNAMES LINKNAME column. This row specifies the TCP/IP host associated with this name translation rule.

     Inbound name translation and come-from checking are not performed for TCP/IP clients.

NEWAUTHID The new end user name (either SNA user ID or DB2 for OS/390 authorization ID). Blank specifies that you do not need to translate the ID.

PASSWORD The password used on the allocate conversation, if passwords are not encrypted (ENCRYPTPSWDS = 'N' in SYSIBM.LUNAMES). If passwords are encrypted, this column is ignored.

6. **SYSIBM.IPNAMES**

   This table is used for TCP/IP nodes.

   LINKNAME The value specified in this column must match the value specified in the LINKNAME column of SYSIBM.LOCATIONS.

   SECURITY_OUT This column defines the DRDA security option that is used when local DB2 SQL applications connect to any remote server associated with this TCP/IP host:

   - **A** signifying the option is "already verified." Outbound connection requests contain an authorization ID and no password. The authorization ID used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending upon the value of the USERNAMES column.
   - **R** signifying the option is "RACF PassTicket." Outbound connection requests contain a userid and a RACF PassTicket. The value specified in the LINKNAME column is used as the RACF PassTicket application name for the remote server.

     The authorization ID used for an outbound request is either the DB2 user's authorization ID or a translated ID, depending upon the value of the USERNAMES column.
   - **P** signifying the option is "password." Outbound connection requests contain an authorization ID and a password. The password is obtained from the SYSIBM.USERNAMES table.

     The USERNAMES column must specify "O."

   USERNAMES This column controls outbound authorization ID translation. Outbound translation is performed when an authorization ID is sent by DB2 to a remote server.

   - **O** signifies an outbound ID is subject to translation. Rows in the SYSIBM.USERNAMES table are used to perform ID translation.

     No translation or "come from" checking is performed on inbound IDs.
   - A blank entry signifies that no translation occurs.

IPADDR This column contains the IP address or domain name of a remote TCP/IP host. The IPADDR column must be specified as follows:

- If the IPADDR contains a left justified character string containing four numeric values delimited by decimal points, DB2 assumes the value is an IP address in dotted decimal format. For example, '123.456.78.91' would be interpreted as a dotted decimal IP address.
- All other values are interpreted as a TCP/IP domain name, which can be resolved by the TCP/IP gethostbyname socket call. TCP/IP domain names are not case sensitive.

Defining Communications (SNA)

VTAM is the Communications Manager for OS/390 systems. VTAM accepts LU 6.2 verbs from DB2 for OS/390 and converts these verbs into LU 6.2 data streams you can transmit over the network. For VTAM to communicate with the partner applications defined in the DB2 for OS/390 CDB, you need to provide VTAM with the following information:

- The LU name for each server.

  When DB2 for OS/390 communicates with VTAM, it is allowed to pass only an LU name (not NETID.LUNAME) to VTAM to identify the desired destination. This LU name must be unique within the LU names known by the local VTAM system, allowing VTAM to determine both the NETID and LU name from the LU name value passed by DB2 for OS/390. When LU names are unique throughout an enterprise's SNA network, it greatly simplifies the VTAM resource definition process. However, this might not always be possible. If LU names within your SNA networks are not unique, you must use VTAM LU name translation to build the correct NETID.LUNAME combination for a nonunique LU name. This process is described in “Resource Name Translation” in the *VTAM Network Implementation Guide*.

  The placement and syntax of the VTAM definitions used to define remote LU names are highly dependent on how the remote system is logically and physically connected to the local VTAM system.
- The RU size, pacing window size, and class of service for each mode name. Create an entry in the VTAM mode table for each mode name specified in the CDB. You also need to define IBMRDB and IBMDB2LM.
- The VTAM and RACF profiles for the LU verification algorithm, if you intend to use partner LU verification.

Setting RU Sizes and Pacing: The VTAM mode table entries you define specify RU sizes and pacing counts. Failure to define these values correctly can have a negative impact on all VTAM applications.

After choosing RU sizes, session limits, and pacing counts, it is extremely important to consider the impact these values can have on the existing VTAM network. You should review the following items when you install a new distributed database system:

- For VTAM CTC connections, verify that the MAXBFRU parameter is large enough to handle your RU size plus the 29 bytes VTAM adds for the SNA request header and transmission header. MAXBFRU is measured in units of 4K bytes, so MAXBFRU must be at least 2 to accommodate a 4K RU.
- For NCP connections, make sure that MAXDATA is large enough to handle your RU size plus 29 bytes. If you specify an RU size of 4K, MAXDATA must be at least 4125.

  If you specify the NCP MAXBFRU parameter, select a value that can accommodate the RU size plus 29 bytes. For NCP, the MAXBFRU parameter defines the number of VTAM I/O buffers that can be used to hold the PIU. If you choose an IOBUF buffer size of 441, MAXBFRU=10 processes a 4K RU correctly because 10*441 is greater than 4096+29.
- *DRDA Connectivity Guide* describes how to assess the impact your distributed database has on the VTAM IOBUF pool. If you use too much of the IOBUF pool resource, VTAM performance is degraded for all VTAM applications.

Defining Communications (TCP/IP)

The considerations are as indicated previously (see “Defining the Local System (TCP/IP)” on page 47).

Provide Security

When a remote system performs distributed database processing on behalf of an SQL application, it must be able to satisfy the security requirements of the Application Requester, the Application Server, and the network connecting them. These requirements fall into one or more of the following categories:

- Selection of end user names
- Network security parameters
- Database manager security
- Security enforced by an external security subsystem
- Data representation

Selecting End User Names

On OS/390 systems, end users are assigned a 1 to 8-character *user ID*. This user ID value must be unique within a particular OS/390 system, but might not be unique throughout the network. For example, there can be a user named JONES on the NEWYORK system, and another user named JONES on the DALLAS system. If these two users are the same person, no conflict exists. However, if the JONES in DALLAS is a different person than the JONES in NEWYORK, the SNA network (and consequently the distributed database systems within that network) cannot distinguish between JONES in NEWYORK and JONES in DALLAS. If you do not correct this situ-

ation, JONES in DALLAS can use the privileges granted to JONES at the NEWYORK system.

To eliminate naming conflicts, DB2 for OS/390 provides support for end user name translation. When an application at the DB2 for OS/390 Application Requester makes a distributed database request, DB2 for OS/390 performs name translation if the communications database specifies that *outbound name translation* is required. If outbound name translation is selected, DB2 for OS/390 always forces a password to be sent with each outbound distributed database request.

Outbound name translation in DB2 for OS/390 is activated by setting the USERNAMES column in the SYSIBM.LUNAMES or SYSIBM.IPNAMES table to either `'O'` or `'B'`. If USERNAMES is set to `'O'`, end user name translation is performed for outbound requests. If USERNAMES is set to `'B'`, end user name translation is performed for both inbound and outbound requests.

Because DB2 for OS/390 authorization is dependent on both the end user's user ID and the user ID of the DB2 for OS/390 plan or package owner, the end user name translation process is performed for the end user's user ID, the plan owner's user ID, and the package owner's user ID.[3] The name translation process searches the SYSIBM.USERNAMES table in the following sequence to find a row that matches one of the following patterns (TYPE.AUTHID.LINKNAME):

1. O.AUTHID.LINKNAME—A translation rule for a specific end user to a specific partner system.
2. O.AUTHID.blank—A translation rule for a specific end user to any partner system.
3. O.blank.LINKNAME—A translation rule for any user to a specific partner system.

If no matching row is found, DB2 for OS/390 rejects the distributed database request. If a row is found, the value in the NEWAUTHID column is used as the authorization ID. (A blank NEWAUTHID value indicates the original name is used without translation.)

Consider the example discussed earlier. You want to give JONES in NEWYORK a different name (NYJONES) when JONES makes distributed database requests to DALLAS. In the example, assume that the application used by JONES is owned by DSNPLAN (the DB2 for OS/390 plan owner), and you do not need to translate this user ID when it is sent to DALLAS. The SQL statements required to supply the name translation rules in the CDB are shown in Figure 19 on page 58.

3 If the request is being sent to a DB2 for OS/390 server, name translation is also performed for the package owner and plan owner. Package and plan owner names never have passwords associated with them.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_OUT, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', 'O');
INSERT INTO SYSIBM.LOCATIONS
     (LOCATION, LINKNAME, LINKATTR)
  VALUES ('DALLAS', 'LUDALLAS', '');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'JONES', 'LUDALLAS', 'NYJONES', 'JONESPWD');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'DSNPLAN', 'LUDALLAS', ' ', 'PLANPWD');
```

Figure 19. SQL for Outbound Name Translation (SNA)

The resulting CDB tables are shown in Figure 20:

NEWYORK.SYSIBM.LOCATIONS

LOCATION	LINKNAME	PORT	TPN
DALLAS	LUDALLAS		

NEWYORK.SYSIBM.LUNAMES

LUNAME	SYSMODENAME	SECURITY-IN	SECURITY-OUT	ENCRYPTPSWDS	MODESELECT	USERNAMES
LUDALLAS			A	N	N	O

NEWYORK.SYSIBM.USERNAMES

TYPE	AUTHID	LINKNAME	NEWAUTHID	PASSWORD
O O	JONES DSNPLAN	LUDALLAS LUDALLAS	NYJONES	JONESPWD PLANPWD

Figure 20. Outbound Name Translation

Figure 21 on page 59 shows a more simple example for connecting to a DB2 Universal Database DRDA AS using an SNA connection.

```
INSERT INTO SYSIBM.LUNAMES (LUNAME,
                            SECURITY_OUT,
                            ENCRYPTPSWDS,
                            USERNAMES)
                      VALUES ('NYX1GW01','P','N','O');
INSERT INTO SYSIBM.LOCATIONS (LOCATION,LINKNAME,TPN)
                           VALUES('TASG6',
                                  'NYX1GW01','NYSERVER');
INSERT INTO SYSIBM.USERNAMES (TYPE,AUTHID,LINKNAME,NEWAUTHID,PASSWORD)
       VALUES ('O','      ','NYX1GW01','SVTDBM6','SG6JOHN');
```

Figure 21. SQL for Outbound Name Translation (simple example for SNA).

Figure 22 on page 60 shows a simple example for connecting to a DB2 Universal Database DRDA AS using a TCP/IP connection.

```
-- DB2 for Solaris1 - UNIX
   DELETE FROM SYSIBM.IPNAMES WHERE LINKNAME = 'SOLARIS1' ;
   INSERT INTO SYSIBM.IPNAMES ( LINKNAME
                              , SECURITY_OUT
                              , USERNAMES
                              , IBMREQD
                              , IPADDR)
                       VALUES ( 'SOLARIS1'
                              , 'P'
                              , 'O'
                              , 'N'
                              , '9.21.45.4')
                              ;
   INSERT INTO SYSIBM.LOCATIONS ( LOCATION
                                , LINKNAME
                                , IBMREQD
                                , PORT
                                , TPN)
                          VALUES ( 'TCPDB1'
                                , 'SOLARIS1'
                                , 'N'
                                , '30088'
                                , '')
                                ;
   INSERT INTO SYSIBM.USERNAMES ( TYPE
                                , AUTHID
                                , LINKNAME
                                , NEWAUTHID
                                , PASSWORD
                                , IBMREQD)
                          VALUES ( 'O'
                                , ''
                                , 'SOLARIS1'
                                , 'svtdbm5'
                                , 'svt5dbm'
                                , 'N')
                                ;
```

Figure 22. SQL for Outbound Name Translation (simple example for TCP/IP).

Network Security

After the Application Requester selects the end user names to represent the remote application, the Application Requester must provide the required network security information.

For SNA connections, LU 6.2 provides three major network security features:

- Session-level security, which is controlled by the VERIFY keyword on the VTAM APPL statement. See the discussion following Figure 17 on page 45 for a description of how to specify session-level security options.

- Conversation-level security, which is controlled by the contents of the SYSIBM.LUNAMES table.
- Data encryption, which is supported only for VTAM 3.4 and later releases of VTAM.

Because the Application Server is responsible for managing the database resources, the Application Server dictates which network security features are required of the Application Requester. You must record the conversation-level security requirements of each Application Server in the SYSIBM.LUNAMES or SYSIBM.IPNAMES tables by setting the USERNAMES column to reflect the application server's requirement.

The possible SNA conversation security options are:

SECURITY=SAME

This is also known as already-verified security because only the end user's user ID is sent to the remote system (no password is transmitted). Use this level of conversation security when the USERNAMES column in SYSIBM.LUNAMES does not contain 'O' or 'B'.

Because DB2 for OS/390 ties end user name translation to outbound conversation security, it does not allow you to use SECURITY=SAME when outbound end user name translation is activated.

SECURITY=PGM

This causes the end user's ID and password to be sent to the remote system for validation. Use this security option when the USERNAMES column of the SYSIBM.LUNAMES table contains either an 'O' or 'B'.

Depending upon options specified in the SYSIBM.LUNAMES table, DB2 for OS/390 obtains the end user's password from two different sources:

- Unencrypted passwords are obtained from the PASSWORD column of the SYSIBM.USERNAMES table. DB2 for OS/390 extracts passwords from the SYSIBM.USERNAMES table when the ENCRYPTPSWDS column in SYSIBM.LUNAMES is not set to 'Y'. Passwords obtained from this source can be transmitted to any DRDA Application Server.

  Figure 23 defines passwords for SMITH and JONES. The LUNAME column in the example contains blanks, so these passwords are used for any remote system SMITH or JONES attempts to access.

```
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'JONES', ' ', ' ', 'JONESPWD');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', 'SMITH', ' ', ' ', 'SMITHPWD');
```

Figure 23. Sending Passwords to Remote Sites (SNA)

- Encrypted passwords are sent to the remote site when the ENCRYPTPSWDS column of SYSIBM.LUNAMES contains 'Y'. Encrypted

passwords are extracted from RACF (or a RACF-equivalent product), and can only be interpreted by another DB2 for OS/390 system. When communicating with a non-DB2 for OS/390 system, do not set ENCRYPTPSWDS to 'Y'.

DB2 for OS/390 searches the SYSIBM.USERNAMES table to determine the user ID (NEWAUTHID value) to transmit to the remote system. This translated name is used for the RACF password extraction. If you do not want to translate names, you must create rows in SYSIBM.USERNAMES that cause names to be sent without translation. Figure 24 allows requests to be sent to LUDALLAS and LUNYC without translating the end user's name (user ID).

```
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', ' ', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('O', ' ', 'LUDALLAS', ' ', ' ');
```

Figure 24. Sending Encrypted Passwords to Remote Sites (SNA)

SECURITY=NONE

This option is not supported by DRDA, so DB2 for OS/390 has no provision for this security option.

Database Manager Security

One way the Application Requester can participate in distributed database security is through outbound name translation, as stated earlier in “Selecting End User Names” on page 56. You can use outbound name translation to control access to each Application Server, based on the identity of the end user making the request and the application making the request. Other ways the DB2 for OS/390 Application Requester contributes to the distributed system security are:

Binding remote applications

End users bind remote applications at the Application Server with the DB2 for OS/390 BIND PACKAGE command. DB2 for OS/390 *does not* restrict the use of the BIND PACKAGE command at the requester. However, an end user cannot use a remote package until the package is included in a DB2 for OS/390 plan. DB2 for OS/390 *does* restrict the use of the BIND PLAN command. An end user cannot add the remote package to a plan unless the end user is given either the BIND or BINDADD privilege with the DB2 for OS/390 GRANT statement.

When you bind a package, use the ENABLE/DISABLE option to specify whether the package is to be used by TSO, CICS/ESA, IMS/ESA, or a remote DB2 for OS/390 subsystem.

Executing remote applications

For the DB2 for OS/390 end user to run a remote application, the end user must have authority to run the DB2 for OS/390 plan associated with that application. The DB2 for OS/390 plan owner automatically has authority to run the plan.

Other end users can be given authority to run the plan with the DB2 for OS/390 GRANT EXECUTE statement. In this way, the owner of a distributed database application can control use of the application on a user-by-user basis.

Security Subsystem

The external security subsystem on OS/390 systems is usually provided by RACF or another product that provide an interface compatible with RACF. The DB2 for OS/390 Application Requester does not have any direct calls to the external security subsystem, with the exception of the encrypted password support described in “Network Security” on page 60. However, the external security subsystem is used indirectly at the Application Requester in the following situations:

- The product responsible for attaching the end user to DB2 for OS/390 uses the external security subsystem to validate the end user's identity (user ID and password). This occurs before the end user is attached to DB2 for OS/390. As stated earlier, CICS/ESA, TSO, and IMS/ESA are examples of products that attach end users to DB2 for OS/390.
- If you use SNA session-level security (via the VERIFY keyword on the DB2 for OS/390 VTAM APPL statement), the external security subsystem is invoked by VTAM to validate the identity of the remote system.

Represent Data

DB2 for OS/390 is shipped with a default installation coded character set identifier (CCSID) of 500. This default is probably *not* correct for your installation.

When installing DB2 for OS/390, you must set the installation CCSID to the CCSID of the characters generated and sent to DB2 for OS/390 by the input devices at your site. This CCSID is generally determined by the national language you use. If the installation CCSID is not correct, character conversion will produce incorrect results. See *DB2 Connect User's Guide* for a list of the supported CCSIDs for each country or national language.

You must ensure that your DB2 for OS/390 subsystem has the ability to convert from each application server's CCSID to your DB2 for OS/390 subsystem's installation CCSID. DB2 for OS/390 provides conversion tables for the most common combinations of source and target CCSIDs, but not for every possible combination. You can add to the set of available conversion tables and conversion routines if you need to. See the *DB2 for OS/390 Version 5 Administration Guide*, for more information about DB2 for OS/390 character conversion.

Setting Up the Application Server

The application server support in DB2 for OS/390 allows DB2 for OS/390 to act as a server for DRDA application requesters. The Application Requester connected to a DB2 for OS/390 Application Server can be:

- A DB2 for OS/390 requester

- A DB2 Universal Database requester, which can run on AIX, HP-UX, OS/2, Solaris, Windows 3.1, Windows 3.11 for Workgroups, Windows 95, or Windows NT. DB2 Connect Enterprise Edition, and DB2 Connect Personal Edition Version 5 provide this function.
- A DB2 Version 2 requester, which can run on AIX, HP-UX, OS/2, Solaris, Windows 3.1, Windows 3.11 for Workgroups, Windows 95, or Windows NT, as wells as Macintosh, SCO, SGI, or SINIX. DDCS Multi-user gateway Version 2.3, DDCS Single-user Version 2.3, and DDCS for Windows Version 2.4 provide this function.
- A OS/400 requester
- An DB2 for VM requester
- Any product that supports the DRDA Application Requester protocols

For any Application Requester connected to a DB2 for OS/390 Application Server, the DB2 for OS/390 Application Server supports database access as follows:

- The Application Requester is permitted to access tables stored at the DB2 for OS/390 application server. The Application Requester must create a package at the DB2 for OS/390 Application Server before the application can be run. The DB2 for OS/390 Application Server uses the package to locate the application's SQL statements at execution time.
- The Application Requester can inform the DB2 for OS/390 Application Server that access must be restricted to read-only activities if the DRDA requester-server connection does not support the two-phase commit process. For example, a DDCS V2R3 requester with a CICS front end would inform the DB2 for OS/390 application server that updates are not allowed.
- The Application Requester can also be permitted to access tables stored at other DB2 for OS/390 systems in the network using system-directed access. System-directed access allows the application requester to establish connections to multiple database systems in a single unit of work.

Provide Network Information

For the DB2 for OS/390 Application Server to properly process distributed database requests, you must take the following steps:

1. Define the application server to the local Communications Manager.
2. Define each potential secondary server destination so the DB2 for OS/390 application server can reroute SQL requests to their final destination.
3. Provide the necessary security.
4. Provide for data representation.

Defining the Application Server (SNA)

For the Application Server to receive distributed database requests, it must be defined to the local Communications Manager and have a unique RDB_NAME. The following discussion relates to SNA connections. You must take the following steps to properly define the Application Server:

1. Select the LU name and RDB_NAME to be used by the DB2 for OS/390 Application Server. The process to record these names in DB2 for OS/390 and VTAM is the same process described in “Defining the Local System (SNA)” on page 42. The RDB_NAME you choose for DB2 for OS/390 must be supplied to all end users and Application Requesters that require connectivity to the Application Server.
2. Register the NETID.LUNAME value for the DB2 for OS/390 Application Server with each Application Requester requiring access, so the Application Requester can route SNA requests to the DB2 for OS/390 server. This is true even in cases where the Application Requester is able to perform dynamic network routing, because the Application Requester must know the NETID.LUNAME before dynamic network routing can be used.
3. Provide the DRDA default TPN (X'07F6C4C2') to each Application Requester because DB2 for OS/390 automatically uses this value.
4. Create an entry in the VTAM mode table for each mode name that is requested by an Application Requester. These entries describe the RU sizes, pacing window size, and class of service for each mode name.
5. Define session limits for the Application Requesters that connect with the DB2 for OS/390 Application Server. The VTAM APPL statement defines default session limits for all partner systems. If you want to establish unique defaults for a particular partner, you can use the SYSIBM.LUMODES table of the communications database (CDB).

   See “Setting RU Sizes and Pacing” on page 55 about how to review your VTAM network.
6. Create entries in the DB2 for OS/390 CDB to identify which Application Requesters are allowed to connect to the DB2 for OS/390 Application Server. Two basic approaches to define the CDB entries for the Application Requesters in the network are:

   a. You can insert a row in SYSIBM.LUNAMES that provides default values to use for any LU not specifically described in the CDB (the default row contains blanks in the LUNAME column). This approach allows you to define specific attributes for some of the LUs in your network, while establishing defaults for all other LUs.

      For example, you can allow the DALLAS system (another DB2 for OS/390 system) to send already-verified distributed database requests (LU 6.2 SECURITY=SAME), while requiring database manager systems to send passwords. Furthermore, you might not want to record an entry in the CDB for each database manager system, especially if there is a large number of these systems. Figure 25 on page 66 shows how the CDB can be used to specify SECURITY=SAME for the DALLAS system, while enforcing SECURITY=PGM for all other requesters.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', ' ');
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES (' ', ' ', 'C', 'N', 'N', ' ');
```

Figure 25. Establishing Defaults for Application Requester Connections (SNA)

b. You can use the CDB to individually authorize each Application Requester in the network, by setting the CDB in one of these ways:

   - Do not record a default row in SYSIBM.LUNAMES. When the default row (the row containing a blank LU name) is not present, DB2 for OS/390 requires a row in SYSIBM.LUNAMES containing the LU name for each application requester that attempts to connect. If the matching row is not found in the CDB, the Application Requester is denied access.
   - Record a default row in SYSIBM.LUNAMES that specifies come-from checking is required (USERNAMES column set to 'I' or 'B'). This causes DB2 for OS/390 to limit access to Application Requesters and end users identified in the SYSIBM.USERNAMES table, as described in “Come-From Checking” on page 67 . You might want to use this approach if your name translation rules require a row with a blank LU name in SYSIBM.LUNAMES, but you do not want DB2 for OS/390 to use this row to allow unrestricted access to the DB2 for OS/390 Application Server.

   In Figure 26, no row contains blanks in the LUNAME column, so DB2 for OS/390 denies access to any LU other than LUDALLAS or LUNYC.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUDALLAS', ' ', 'A', 'N', 'N', ' ');
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS, MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', ' ');
```

Figure 26. Identifying Individual Application Requester Connections (SNA)

Defining the Application Server (TCP/IP)

For the Application Server to receive distributed database requests over TCP/IP connections, it must be defined to the local TCP/IP subsystem, and have a unique RDB_NAME. Additionally, the DB2 for OS/390 Bootstrap Dataset must include the necessary parameters, and you may need to make updates to the DB2 for OS/390 Communications Database (CDB).

1. For information about setting up TCP/IP at the AS, see *DB2 for OS/390 Installation Reference*. How to set up the AR is described in *DB2 Connect Enterprise Edition Quick Beginnings*, and *DB2 Connect Personal Edition Quick Beginnings*.
2. An example Bootstrap Dataset definition is shown in Figure 18 on page 48.
3. No CDB updates are required if you will only use inbound database connections, so that if you plan to use DB2 for OS/390 only as a server, you do not need to populate the CDB, and default values can be used. A simple example of how to update SYSIBM.IPNAMES follows.

   If you want to permit inbound database connection requests for TCP/IP nodes, you can use an SQL command such as the following to update this table:

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME) VALUES('        ')
```

Provide Security

When an Application Requester routes a distributed database request to the DB2 for OS/390 Application Server, the following security considerations can be involved:

- Come-from checking
- Selection of end user names
- Network security parameters
- Database manager security
- Security enforced by an external security subsystem

Come-From Checking

When the DB2 for OS/390 Application Server receives an end user name from the Application Requester, the Application Server can restrict the end user names received from a given Application Requester. This is accomplished through the use of *come-from* checking. Come-from checking allows the Application Server to specify that a given user ID is only allowed to be used by particular partners. For example, the Application Server can restrict JONES to “come from” DALLAS. If another Application Requester (other than DALLAS) attempts to send the name JONES to the Application Server, the Application Server can reject the request because the name did not come from the correct network location.

DB2 for OS/390 implements come-from checking as part of inbound end user name translation, which is described in the next section.

Note: Inbound and come-from checks are not done for TCP/IP inbound requests.

Selecting End User Names

The user ID passed by the Application Requester might not be unique throughout the entire SNA network. The DB2 for OS/390 Application Server might need to perform inbound name translation to create unique end user names throughout the SNA network. Similarly, the DB2 for OS/390 Application Server might need to perform out-bound name translation to provide a unique end user name to the secondary servers

involved in the application (see “Provide Security” on page 56 for information concerning outbound end user name translation).

Inbound name translation is enabled by setting the USERNAMES column of the SYSIBM.LUNAMES or SYSIBM.IPNAMES table to `'I'` (inbound translation) or `'B'` (both inbound and outbound translation). When inbound name translation is in effect, DB2 for OS/390 translates the user ID sent by the Application Requester and the DB2 for OS/390 plan owner's name (if the Application Requester is another DB2 for OS/390 system).

If the Application Requester sends both a user ID and a password on the APPC ALLOCATE verb, the user ID and password are validated before the user ID is translated. The PASSWORD column in SYSIBM.USERNAMES is not used for password validation. Instead, the user ID and password are presented to the external security system (RACF or a RACF-equivalent product) for validation.

When the incoming user ID on the ALLOCATE verb is verified, DB2 for OS/390 has authorization exits you can use to provide a list of secondary AUTHIDs and perform additional security checks. See the *DB2 for OS/390 Version 5 Administration Guide*, for details.

The inbound name translation process searches for a row in the SYSIBM.USERNAMES table, which must fit one of the patterns shown in the following precedence list (TYPE.AUTHID.LINKNAME):

1. I.AUTHID.LINKNAME—A specific end user from a specific Application Requester
2. I.AUTHID.blank—A specific end user from any Application Requester
3. I.blank.LINKNAME—Any end user from a specific Application Requester

If no row is found, remote access is denied. If a row is found, remote access is allowed and the end user's name is changed to the value provided in the NEWAUTHID column, with a blank NEWAUTHID value indicating that the name is unchanged. Any DB2 for OS/390 resource authorization checks (for example, SQL table privileges) made by DB2 for OS/390 are performed on the translated end user names, rather than on the original user names.

When the DB2 for OS/390 Application Server receives an end user name from the Application Requester, several objectives can be accomplished by using the DB2 for OS/390 inbound name translation capability:

- You can change an end user's name to make it unique. For example, the following SQL statements translate the end user name JONES from the NEWYORK application requester (LUNAME LUNYC) to a different name (NYJONES).

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'JONES', 'LUNYC', 'NYJONES', ' ');
```

- You can change the end user's name so that a group of end users are all represented by a single name. For example, you might want to represent all users from the NEWYORK Application Requester (LUNAME LUNYC) with the user name NYUSER. This allows you to grant SQL privileges to the name NYUSER and to control the SQL access given to users from NEWYORK.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUNYC', 'NYUSER', ' ');
```

- You can restrict the end user names transmitted by a particular Application Requester. This use of end user name translation accomplishes the come-from check described in “Come-From Checking” on page 67. For example, the SQL statements that follow allow only SMITH and JONES as end user names from the NEWYORK Application Requester. Any other name is denied access, because it is not listed in the SYSIBM.USERNAMES table.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES ('LUNYC', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'SMITH', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', 'JONES', 'LUNYC', ' ', ' ');
```

- You can restrict the Application Requesters allowed to connect to the DB2 for OS/390 Application Server. This is yet another feature of come-from checking. The following example accepts any end user name sent by the NEWYORK Application Requester (LUNYC) or the CHICAGO Application Requester (LUCHI). Other Application Requesters are denied access, because the default SYSIBM.LUNAMES row specifies inbound name translation for all inbound requests.

```
INSERT INTO SYSIBM.LUNAMES
     (LUNAME, SYSMODENAME, SECURITY_IN, ENCRYPTPSWDS,
              MODESELECT, USERNAMES)
  VALUES (' ', ' ', 'A', 'N', 'N', 'I');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUNYC', ' ', ' ');
INSERT INTO SYSIBM.USERNAMES
     (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
  VALUES ('I', ' ', 'LUCHI', ' ', ' ');
```

Provide Network Security

For SNA connections LU 6.2 provides three major network security features:

- Session-level security
- Conversation-level security
- Encryption

"Network Security" on page 60 discusses how to specify session-level security and encryption with DB2 for OS/390. The DB2 for OS/390 Application Server uses session-level security and encryption in exactly the same manner as the DB2 for OS/390 Application Requester.

The only remaining network security consideration is SNA conversation-level security. Some aspects of conversation-level security are unique for a DB2 for OS/390 Application Server. The DB2 for OS/390 Application Server plays two distinct roles in network security:

- As a requester to secondary servers, the DB2 for OS/390 Application Server is responsible for issuing APPC requests that contain the SNA conversation-level security parameters required by the secondary servers. The DB2 for OS/390 Application Server uses the USERNAMES column of the SYSIBM.LUNAMES table and the SYSIBM.USERNAMES table to define the SNA conversation level security requirements for each secondary server. The details of these definitions are identical to those in "Network Security" on page 60.
- As the server for the Application Requester, the DB2 for OS/390 Application Server dictates the SNA conversation level security requirements for the Application Requester. DB2 for OS/390 uses the USERSECURITY column of the SYSIBM.LUNAMES table to determine the conversation security required from each Application Requester in the network. The following values are used in the USERSECURITY column:

  C

  This indicates that DB2 for OS/390 requires the Application Requester to send a user ID and password (LU 6.2 SECURITY=PGM) with each distributed database request. If the ENCRYPTPSWDS column in SYSIBM.LUNAMES contains 'Y', DB2 for OS/390 assumes the password is already in RACF encrypted format (this is only possible for a DB2 for OS/390 Application Requester). If the ENCRYPTPSWDS column does not contain

'Y', DB2 for OS/390 expects the password in the standard LU 6.2 format (EBCDIC character representation). In either case, DB2 for OS/390 passes the user ID and password values to the security subsystem for validation. You must have a security subsystem that provides APPC user ID and password verification; for example, RACF has the capability to verify APPC user IDs and passwords. If the security subsystem rejects the user ID-password pair, distributed database access is denied.

Any other value
This indicates the Application Requester is allowed to send either an already-verified user ID (LU 6.2 SECURITY=SAME) or a user ID and password (LU 6.2 SECURITY=PGM). If a user ID and password are sent, DB2 for OS/390 processes them as described for 'C' above. If the request contains only a user ID, the security subsystem is called to authenticate the user unless the SYSUSERNAMES table is used to manage inbound user IDs.

If a security violation is discovered, LU 6.2 requires the DB2 for OS/390 Application Server to return the SNA security failure sense code ('080F6051'X) to the Application Requester. Because this sense code does not describe the cause of the failure, DB2 for OS/390 provides two methods for recording the cause of distributed security violations:

- A DSNL030I message is produced, which provides the requester's LUWID and a DB2 reason code describing the failure. DSNL030I also includes the AUTHID, if known, that was sent from the application request that was rejected.
- An alert is recorded in the NETVIEW hardware monitor database, which contains the same information provided in the DSNL030I message.

Database Manager Security

As the owner of database resources, the DB2 for OS/390 Application Server controls the database security functions for SQL objects residing at the DB2 for OS/390 Application Server. Access to DB2 for OS/390-managed objects is controlled by privileges, which are granted to users by the DB2 for OS/390 administrator or the owners of individual objects. The two basic classes of objects that the DB2 for OS/390 Application Server controls are:

- **Packages**— Individual end users are authorized to create, replace, and run packages with the DB2 for OS/390 GRANT statement. When an end user owns a package, that user can automatically run or replace the package. Other end users must be specifically authorized to run a package at the DB2 for OS/390 Application Server with the GRANT statement. USE can be granted to individual end users or to PUBLIC, which allows all end users to run the package.

  When an application is bound to DB2 for OS/390, the package contains the SQL statements contained in the application program. These SQL statements are classified as:

  Static SQL
  Static SQL means that the SQL statement and the SQL objects referenced by the statement are known at the time the application is bound to DB2 for

OS/390. The person creating the package must have authority to execute each of the static SQL statements contained in the package.

When end users are granted authority to execute a package, they automatically have authority to execute each of the static SQL statements contained in the package. Thus, end users do not need any DB2 for OS/390 table privileges if the package they execute contains only static SQL statements.

Dynamic SQL

Dynamic SQL describes an SQL statement that is not known until the program executes. In other words, the SQL statement is built by the program and dynamically bound to DB2 for OS/390 with the SQL PREPARE statement. When an end user executes a dynamic SQL statement, the user must have the table privileges required to execute the SQL statement. Because the SQL statement is not known at the time the plan or package is created, the end user is not automatically given the required authority by the package owner.

- **SQL objects**— These are tables, views, synonyms, or aliases. DB2 for OS/390 users can be granted various levels of authority to create, delete, change, or read individual SQL objects. This authority is required to bind static SQL statements or to execute dynamic SQL statements.

When you create a package, the DISABLE/ENABLEoption allows you to control which DB2 for OS/390 connection types can run the package. You can use RACF and DB2 for OS/390 security exit routines to selectively allow end users to use DDF. You can use RLF to specify limits on processor time for remote binds and dynamic SQL executions.

Consider a DB2 for OS/390 package named MYPKG, which is owned by JOE. JOE can allow SAL to execute the package by issuing the DB2 for OS/390 GRANT USE statement. When SAL executes the package, the following occurs:

- DB2 for OS/390 verifies that SAL was given USE authority for the package.
- SAL can issue every static SQL statement in the package because JOE had the required SQL object privileges to create the package.
- If the package has dynamic SQL statements, SAL must have SQL table privileges of her own. For example, SAL cannot issue SELECT * FROM JOE.TABLE5 unless she is granted read access to JOE.TABLE5.

Security Subsystem

The DB2 for OS/390 Application Server use of the security subsystem (RACF or a RACF-equivalent product) is dependent on how you define the inbound name translation function in the SYSIBM.LUNAMES table:

- If you specify 'I' or 'B' for the USERNAMES column, inbound name translation is active, and DB2 for OS/390 assumes that the DB2 for OS/390 administrator is using inbound name translation to perform part of the system security enforcement. The external security subsystem is called only if the Application Requester sends a request containing both user ID and password (SECURITY=PGM). You must have

a security subsystem that provides APPC user ID and password verification; for example, RACF has the capability to verify APPC user IDs and passwords.

If the request from the Application Requester contains only a user ID (SECURITY=SAME), the external security system is not called at all, because the inbound name translation rules define which users are allowed to connect to the DB2 for OS/390 Application Server.

- If you specify something other than 'I' or 'B' for the USERNAMES column, the following security subsystem checks are performed:
  - When a distributed database request is received from the Application Requester, DB2 for OS/390 calls the external security system to validate the end user's user ID (and password if it is provided).
  - The external security system is called to verify that the end user is authorized to connect to the DB2 for OS/390 subsystem.
- In either case, an authorization exit is driven to provide a list of secondary authorization IDs. For more information, see the *DB2 for OS/390 Version 5 Administration Guide*.

Represent Data

You must ensure that your DB2 for OS/390 subsystem has the ability to convert from each application requester's CCSID to your DB2 for OS/390 subsystem's installation CCSID. Refer to “Represent Data” on page 63 for more information.

Chapter 3. Connecting DB2 for AS/400 in a DRDA Network

OS/400 contains DB2 for AS/400, the IBM relational database management system for AS/400 systems.

The following material explains how to connect an AS/400 system to a distributed relational database. The emphasis in this chapter is on connecting an AS/400 system to unlike DRDA systems. For information on connecting two AS/400 systems, see *AS/400 Distributed Database Programming*.

DB2 for AS/400 Implementation

This chapter describes how DB2 for AS/400 provides support for distributed database systems. The OS/400 Version 2 Release 1 Modification 1 licensed program supported DRDA remote unit of work, and OS/400 Version 3 Release 1 added support for DRDA distributed unit of work (DUOW). This support is part of the OS/400 operating system. This means you do not need the DB2 for AS/400 Query Manager and SQL Development Kit licensed programs to use the DRDA support or to run programs with embedded SQL statements.

Setting Up the Application Requester

The AS/400 system implements the DRDA application requester support as an integral part of the OS/400 operating system. Because application requester support is part of the OS/400 operating system, it is active whenever the operating system is active. This is also true of the application server support in DB2 for AS/400.

When DB2 for AS/400 acts as an application requester, it can connect to any application server that supports DRDA. For the DB2 for AS/400 application requester to provide distributed database access, you need to consider the following:

- Providing network information
- Providing security
- Representing Data

Provide Network Information

The Application Requester must be able to accept a relational database name and translate it into network parameters. The AS/400 system uses the relational database directory to register relational database names and their corresponding network parameters. This directory allows the AS/400 application requester to pass the required network information to establish communications in a distributed database network.

Much of the processing in a distributed database environment requires messages to be exchanged with other locations in the network. For this processing to be performed correctly in the SNA environment you need to do the following:

Define the local system to DB2 for AS/400

Define the remote system to DB2 for AS/400
Define communications to DB2 for AS/400

Defining the Local System to DB2 for AS/400

Each application requester in the distributed database network must have an entry in its Relational Database Directory for its local relational database and one for each remote relational database the application requester accesses. Any AS/400 system in the distributed database network that acts only as an application server must have an entry in its relational database directory for the local relational database. For more information about the relational database directory, see *AS/400 Distributed Database Programming*.

To define the local system, name the local database by adding an entry with a remote location name of *LOCAL to the relational database directory. To do this, use the Add Relational Database Directory Entry (ADDRDBDIRE) command. The following example shows the ADDRDBDIRE command, where the name of the application requester's database is ROCHESTERDB:

```
ADDRDBDIRE RDB(ROCHESTERDB) RMTLOCNAME(*LOCAL)
```

For more detail on relational database directory commands, see *AS/400 Distributed Database Programming*.

Defining the Remote System to DB2 for AS/400

Each application server in the distributed DB network must also have a local entry in its RDB directory. In addition, an entry for each remote database must be present in the RDB directory of each application requester. To create these:

- Define the remote databases to the local database by adding an entry for each remote database in the relational database directory using the ADDRDBDIRE or WRKRDBDIRE command. For SNA communications the information you can specify includes:
  - Remote database name
  - Remote location name of the database
  - Local location name
  - Mode name used to establish the communications
  - Remote network identifier
  - Name of the device used for the communications
  - Transaction program name of the remote database

  For most cases, the only information needed is the remote database name and the remote location name[4] of the database. When only the remote location name is specified, default values are used for the remaining parameters. The system selects a device description using the remote location name.

  If more than one device description contains the same remote location name and a specific device description is required, then the values for local location name and

4 "Location name" in OS/400 is synonymous with "LU name" in VTAM. "Remote location name" means "partner or remote LU name."

remote network identifier in the relational database directory entry should match the values in the device description. The selection of device descriptions can be complicated if the same remote location name is used in more than one device description. Use unique remote location names in each device description to avoid confusion. The transaction program name of the remote database defaults to the DRDA default transaction program name of X'07F6C4C2'.

The communications information in the relational database directory is used to establish a conversation with the remote system.

For TCP/IP connections (when supported), only the remote database name and associated IP address and port are required.

Defining SNA Communications

This section describes configuring communications on the AS/400 system using Advanced Peer-to-Peer Networking (APPN). The AS/400 system also allows advanced program-to-program communications (APPC) configurations, which do not provide network routing support. An AS/400 distributed database works with either configuration. For more information about APPC configurations, see *OS/400 Communications Configuration*.

AnyNet Support on the AS/400 allows APPC applications to run over Transmission Control Protocol/Internet Protocol (TCP/IP) networks. Examples in the sections that follow include DDM, Systems Network Architecture Distribution Services, Alerts, and 5250 Display Station Pass-Through. These applications, along with DRDA, can run unchanged over TCP/IP networks with some additional configuration. To specify AnyNet support, you specify *ANYNW on the LINKTYPE parameter of the CRTCTLAPPC command.

For more information on APPC over TCP/IP, refer to *OS/400 Communications Configuration* and *OS/400 TCP/IP Configuration and Reference*. Availability of native TCP/IP support for DRDA is expected during 1998.

APPN provides networking support that allows the AS/400 system to participate in and control a network of systems without requiring the networking support traditionally provided by a mainframe system. The following steps tell you how to configure an AS/400 system for APPN support.

1. Define the network attributes using the Change Network Attributes (CHGNETA) command.

   The network attributes contain:

   - The local system name
   - The name of the system in the APPN network
   - The local network identifier
   - The network node type
   - The names of the network servers used by the AS/400 system, if the machine is an end node
   - The network control points, if the AS/400 is an end node

2. Create the line description.

   The line description describes the physical line connection and the data link protocol to be used between the AS/400 system and the network. Use the following commands to create line descriptions:

   - Create line description (Ethernet) (CRTLINETH)
   - Create line description (SDLC) (CRTLINSDLC)
   - Create line description (token ring) (CRTLINTRN)
   - Create line description (X.25) (CRTLINX25)

3. Create controller descriptions.

   The controller description describes the adjacent systems in the network. Indicate the use of APPN support by specifying APPN(*YES) when creating the controller description. Use the following commands to create controller descriptions:

   - Create controller description (APPC) (CRTCTLAPPC)
   - Create controller description (SNA HOST) (CRTCTLHOST)

   If the AUTOCRTCTL parameter on a token-ring or Ethernet line description is set to *YES, then a controller description is automatically created when the system receives a session start request over the token-ring or Ethernet line.

4. Create a class-of-service description.

   Use class-of-service description to select the communication routes (transmission groups) and give transmission priority. Five class-of-service descriptions are supplied by the system:

   #CONNECT The default class of service.

   #BATCH A class of service for batch jobs.

   #BATCHSC The same as #BATCH except a data link security of at least a packet-switched network is required. In packet-switched networks, data does not always follow the same path through the network.

   #INTER A class of service tailored for interactive communications.

   #INTERSC The same as #INTER except a data link security of at least a packet-switched network is required.

   Create other class-of-service descriptions using the Create Class-of-Service (CRTCOSD) command.

5. Create a mode description.

   The mode description gives the session characteristics and number of sessions that can be used to negotiate the allowed values between the local and remote location. The mode description also points to the class of service that is used for the conversation. Several predefined modes are shipped with the system:

   BLANK The default mode name specified in the network attributes when the system is shipped.

   #BATCH A mode tailored for batch jobs.

#BATCHSC The same as #BATCH except the associated class-of-service description requires a data link security of at least a packet-switched network.

#INTER A mode tailored for interactive communications.

#INTERSC The same as #INTER except the associated class-of-service description requires a data link security of at least a packet-switched network.

IBMRDB A mode tailored for DRDA communications.

Other mode descriptions can be created using the Create Mode Description (CRTMODD) command.

6. Create device descriptions.

   The device description provides the characteristics of the logical connection between the local and remote systems. You do not have to manually create device descriptions if the AS/400 system is running to a host system with APPN and as an independent logical unit (LU). The AS/400 system automatically creates the device description and attaches it to the appropriate controller description when the session is established. If the AS/400 system is a dependent LU, then you must manually create the device descriptions using the Create Device Description (CRTDEVAPPC) command. In the device description, specify APPN(*YES) to indicate that the APPN is being used.

7. Create APPN location lists.

   If additional local locations (called *LUs* on other systems) or special characteristics of remote locations for APPN are required, then you need to create APPN location lists. The local location name is the control point name specified in the network attributes. If you need additional locations for the AS/400 system, an APPN local location list is required. An example of a special characteristic of a remote location is if the remote location is in a network other than the one the local location is in. If the conditions exist, an APPN remote location list is required. Create APPN location lists by using the Create Configuration List (CRTCFGL) command.

8. Activate (vary on) communications.

   You can activate the communication descriptions by using the Vary Configuration (VRYCFG) command or the Work With Configuration Status (WRKCFGSTS) command. If the line descriptions are activated, then the appropriate controllers and devices attached to that line are also activated. The WRKCFGSTS command is also useful for viewing the status of each connection.

9. RU sizes and pacing

   RU sizes and pacing are controlled by values specified in the mode description. When you create the mode description, defaults are provided for both RU size and pacing. The default values are an AS/400 estimate for most environments including a distributed database. If the default is taken for RU size, the AS/400 system estimates the best value to use. When the AS/400 system is communicating with another system that supports adaptive pacing, the pacing values specified are only a starting point. The pacing is adjusted by each system depending on the system's

ability to handle the data sent to it. For systems that do not support adaptive pacing, the pacing values are negotiated at session start, and remain the same for the life of the session. For more information, see *OS/400 Communications Configuration*.

Notes:

1. The controller description is equivalent to the IBM Network Control Program and Virtual Telecommunications Access Method (NCP/VTAM) physical unit (PU) macros.
2. The device description is equivalent to the NCP/VTAM logical unit (LU) macro. The device description contains information similar to that stored in the Communications Manager/2 1.1 partner LU profile.
3. The mode description is equivalent to the NCP/VTAM mode tables and the Communications Manager Transmission Service Mode profile.

For more information about configuring for networking support and working with location lists, see the *OS/400 Communications Configuration* and *APPN Support*. For examples showing use of CL commands to define system configurations, see *AS/400 Distributed Database Programming*.

Provide Security

When a remote system performs distributed database processing on behalf of an SQL application, it must be able to satisfy the security requirements of the application requester, the application server, and the network connecting them. These requirements fall into one or more of the categories that follow:

- Selection of end user names
- Network security parameters
- Database manager security
- Security enforced by AS/400 security

Selecting End User Names

On AS/400 systems, end users are assigned a 1- to 10-character user ID that is unique to that system, but not necessarily unique within the network. This user ID is the one passed to the remote system when the connection is being established between two databases. To avoid conflicts between user IDs on systems in the network, outbound name translation is often used to change the user ID to resolve the conflict before it is sent over the network. However, the AS/400 system does not provide any outbound name translation to resolve potential conflicts at the server. These conflicts must be resolved at the application server, unless you use the additional USER and USING clauses on the AS/400 SQL CONNECT statement. USER is a valid ID on the application server, and USING is the corresponding password for the user.

Network Security

After the application requester selects the end user names to represent the remote application, it must provide the required LU 6.2 network security information. LU 6.2 provides three major network security features:

- Session-level security, controlled by the LOCPWD keyword on the CRTDEVAPPC command
- Conversation-level security, controlled by the OS/400 operating system
- Encryption, not supported by the OS/400 operating system

Session-level security is provided through LU-to-LU verification. Each LU has a key that must match the key at the remote LU. You specify the key on the LOCPWD keyword on the CRTDEVAPPC command.

Because the application server is responsible for managing the database resources, the application server dictates which network security features are required of the application requester. The AS/400 security administrator must verify the security requirements of each application server so they require no more than the AS/400 application requester supports.

The following are possible SNA conversation security options:

SECURITY=SAME
: Also known as already-verified security. Only the user ID of an application user is sent to the remote system. No password is sent. Before the AS/400 Version 2 Release 2 Modification 0, this level of conversation security was the only level supported by an AS/400 application requester.

SECURITY=PGM
: Causes both the user ID and the password of the application user to be sent to the remote system for validation. Before the AS/400 Version 2 Release 2 Modification 0, this security option was not supported by an AS/400 application requester.

SECURITY=NONE
: Not supported when AS/400 is an application requester.

Database Manager Security

The AS/400 system does not have an external security subsystem. All security is handled through the OS/400 operating system as discussed in the following section, “System Security.”

System Security

The OS/400 operating system controls authorization to all objects on the system, including programs, packages, tables, views, and collections.

The application requester controls authorization to objects that reside on the application requester. The security for objects on the application server is controlled at the application server, on the basis of which user ID is sent from the application requester. The user ID sent to the application server is associated with the user of the AS/400 application requester or the user ID given in the USER clause of the AS/400 SQL CONNECT statement. For example, CONNECT TO rdbname USER userid USING password.

Security of objects can be managed using the object authority CL commands or with the SQL statements GRANT and REVOKE. The object CL authority commands include Grant Object Authority (GRTOBJAUT) and Revoke Object Authority (RVKOBJAUT). These commands work on any object on the system. The statements GRANT and REVOKE only work on SQL objects: tables, views, and packages. If you need to change authorization for other objects such as programs or collections, use the GRTOBJAUT and RVKOBJAUT commands.

Granting and Revoking Authority: Enter the following command on an AS/400 system to grant *USE authority to user USER1 to program PGMA:

```
GRTOBJAUT OBJ(PGMA) OBJTYPE(*PGM) USER(USER1) AUT(*USE)
```

The command to revoke the same authority:

```
RVKOBJAUT OBJ(PGMA) OBJTYPE(*PGM) USER(USER1) AUT(*USE)
```

*PGM identifies the object type in this example as a program. *SQLPKG is used to operate on a package, *LIB is used for a collection, and *FILE is used for a table.

GRTOBJAUT and RVKOBJAUT can also be used to prevent users from creating programs and packages. When authority is revoked from any of the CRTSQLxxx commands (where xxx = RPG, C, CBL, FTN, or PLI) used to create programs, a user is not able to create programs. If authority is revoked to the CRTSQLPKG command, the user is not able to create packages from the application requester or on the application server.

For example, enter the following command on an AS/400 system to grant *USE authority to user USER1 to the CRTSQLPKG command:

```
GRTOBJAUT OBJ(CRTSQLPKG) OBJTYPE(*CMD) USER(USER1) AUT(*USE)
```

This affects the execution of crtsqlpkg on the application requester. On the application server, this command allows the creation of packages.

The command to revoke the same authority is:

```
RVKOBJAUT OBJ(CRTSQLPKG) OBJTYPE(*CMD) USER(USER1) AUT(*USE)
```

Applying Default Authorization: When objects are created, they are given a default authorization. By default, the creator of a table, view, or program is given all authority on those objects. Also by default, the public is given the same authority on those objects as they (the public) have on the objects' library or collection.

For more information on system security see *AS/400 Security - Reference*.

Represent Data

Products supporting DRDA automatically perform any necessary conversions at the receiving system. For this to happen the application requester CCSID value must be a supported value for conversion by the receiving system.

On an application requester, you should be concerned with the CCSID associated with:

- Requesting Job

  OS/400 work management support initializes the job CCSID to the CCSID specified on the user profile. If the user profile CCSID value is *SYSVAL, work management support gets the CCSID from the QCCSID system value. The system value QCCSID is initially set to CCSID 65535. The use of 65535 for the CCSID of jobs servicing connect attempts from DB2 Universal Database will cause the connection attempts to fail. Changing the system value QCCSID affects the whole system, so the recommended action is to change the CCSID of the user profile for the job under which the server job runs. Set the CCSID of the user profile for the job to an appropriate value. For example, use CCSID 37 for US English. In general, the the appropriate choice would be to use the default coded character set identifier for the AS/400 you are connecting to.

  The job CCSID can be changed by using the Change Job (CHGJOB) command. Or for subsequent jobs use the Change User Profile (CHGUSRPRF) command to change the CCSID value of the user profile. To see what CCSID is in effect for a job, in a CL program, use the Retrieve Job Attributes (RTVJOBA) command to get the current job CCSID. Interactively, use the Work with Job (WRKJOB) command and select option 2, Display Job Definition Attributes on the Work with Job display.

- Database Physical Files

  Database physical files default to the default job CCSID (which may be different from the job CCSID) at file creation if a CCSID is not explicitly specified on the Create Physical File (CRTPF) or Create Source Physical File (CRTSRCPF) command. Prior to DB2 for AS/400 V3R1, the default was the job CCSID which was often 65535 and inappropriate for DRDA usage. The default job CCSID is never 65535, and it is therefore a better choice for the CCSID of physical files accessed via DRDA.

  You can use the Display File Description (DSPFD) command to view the CCSID of a file or the Display File Field Description (DSPFFD) command to view the CCSID of the fields of a file.

  Use the Change Physical File (CHGPF) command to change the CCSID of a physical file. A physical file cannot always be changed if one or more of the following conditions exist:

  – Logical files are defined over the physical file. In this case you may need to do the following:

    1. Save the logical and physical files along with their access paths.
    2. Print a list of authorities for logical files (DSPOBJAUT).
    3. Delete the logical files.
    4. Change the physical files.
    5. Restore the physical and logical files and their access paths over the changed physical files.
    6. Grant private authority to the logical files (see the list that you printed).

  – Files or fields are explicitly assigned a CCSID value. To change a physical file with the CCSID assigned at the field level, recreate the physical file and copy

the data to the new file using the FMTOPT(*MAP) parameter on the Copy File (CPYF) command.

– Record formats are being shared in a version of OS/400 prior to Version 3 Release 1.

Setting Up the Application Server

The application server support on the AS/400 system allows it to act as a server for DRDA application requesters. The application requester connected to a DB2 for AS/400 application server can be any client that supports DRDA protocols.

The application requester is permitted to access tables stored locally at the DB2 for AS/400 application server. The application requester must create a package at the DB2 for AS/400 application server before any SQL statements can be run. The DB2 for AS/400 application server uses the package containing the application's SQL statements at program run time.

Provide Network Information

To process distributed database requests on the AS/400 application server, you need to name the application server database in the RDB directory. For SNA Communications you need to define the application server system, and set the request and response unit sizes and pacing.

Naming the Application Server Database

You name the application server database (at the application server location) in the same way that you identify the application requester database (at the application requester location). Use the Add Relational Database Directory Entry (ADDRDBDIRE) command, and specify *LOCAL as the remote location.

Defining the Application Server to the Network

For access using SNA, defining the application server to the network is identical to defining the application requester to the network. You need to create line, controller, device, and mode descriptions to define both the application server and the application requester that sends the requests. For information on how to define the application server to the network, see “Defining the Local System to DB2 for AS/400” on page 76 and “Defining the Remote System to DB2 for AS/400” on page 76. See also *AS/400 Distributed Database Programming*.

The transaction program name used to start an AS/400 application server database is the DRDA default X'07F6C4C2'. This transaction program name is defined within the AS/400 system to start the application server. The corresponding parameter for TCP/IP connections, when that protocol is supported by DB2/400, is the port. DB2/400 will always use the DRDA well-known port of 446 as a server.

Setting RU Sizes and Pacing

Network definitions must be reviewed to determine if the distributed database network impacts the existing network. These considerations are the same for the application server and the application requester.

Provide Security

When an application requester routes a distributed database request to the AS/400 application server, the following security considerations can be involved:

- Selection of end user names
- Network security parameters
- Database manager security
- AS/400 security

Selecting End User Names

The application requester sends a user ID to the application server for security processing. The job running on the AS/400 application server uses this user ID, or in some instances, a default user ID.

The AS/400 application server does not provide inbound user ID translation to resolve conflicts among user IDs that are not unique or group multiple users under a single user ID. Each user ID sent from an Application Requester must exist on the application server. A method to group incoming requests into a single user ID, with loss of some security, is to specify a default user ID in a communications entry in the subsystem that is handling the remote job start requests. See the descriptions of `ADDCMNE` and `CHGCMNE` in the *AS/400 CL Reference*.

SNA Network Security

LU 6.2 provides three major network security features:

- Session-level security
- Conversation-level security
- Encryption (not supported by the AS/400 system)

The DB2 for AS/400 application server uses session-level security in exactly the same manner as the DB2 for AS/400 application requester.

The application server controls the SNA conversation levels used for the conversation. The SECURELOC parameter on the APPC device description or the secure location value on the APPN remote location list determines what is accepted from the application requester for the conversation.

The possible SNA conversation security options are:

SECURITY=SAME

Also known as already-verified security. Only the user ID of the application user is required by the application server. No password is sent. Use this level of conversation security at the application server by setting the

SECURELOC parameter on the APPC device description to *YES or by setting the secure location value on the APPN remote location list to *YES.

SECURITY=PGM

Causes both the user ID and password to be required by the application server for validation. Use this level of conversation security at the application server by setting the default user ID in the AS/400 subsystem communications entry to *NONE (no default user ID) and by setting the SECURELOC parameter or the secure location value to *NO.

SECURITY=NONE

An application server does not expect a user ID or password. The conversation is allowed using a default user profile on the application server. To use this option, specify a default user profile in the subsystem communications directory and specify *NO for the SECURELOC parameter or the secure location value.

SNA/DS (SNA Distribution Services) requires a default user ID, so SNA/DS should have its own subsystem for the normal case where you don't want a default user id for DRDA applications.

A method for grouping incoming start job requests into a single user ID was mentioned in “Selecting End User Names” on page 85. This method does not verify the user ID sent from the Application Requester. The application server job is started under a default user ID, and the user who initiated the connection from the application server has access at the application server even if the user ID sent has restricted authorization. This is done by defining the application server as a nonsecure location, specifying a default user ID in the AS/400 subsystem communications entry, and configuring the application requester to send a user ID only during connection processing. If a password is sent, the user ID that accompanies it is used instead of the default user ID.

The AS/400 subsystem communications entries are distinguished by the device and mode name used to start the conversation. By assigning different default user IDs to different device/mode pairs, users can be grouped by how they are communicating with the application server.

The AS/400 system also offers a network security feature that is used only for distributed database and distributed file management. A network attribute for these types of system access exists that either rejects all attempts to access or allows the security to be controlled by the system on an object-by-object basis.

TCP/IP Network Security

A new command named CRTDDMTCPA will be available in future releases of OS/400 to allow for specifying whether a server will accept TCP/IP connect requests without a password.

Database Manager Security

All security is handled through the OS/400 security function.

System Security

The AS/400 system does not have an external security subsystem. All security is handled by the OS/400 security function that is an integral part of the operating system. The operating system controls authorization to all objects on the system, including programs, packages, tables, views, and collections.

The application server controls authorizations to the objects that reside on the application server. The security control for those objects is based on which user ID starts the application server job. This user ID is determined as described in “Selecting End User Names” on page 85.

Security of objects can be managed through the use of the object authority CL commands or through the SQL statements GRANT and REVOKE. The object authority CL commands include Grant Object Authority (GRTOBJAUT) and Revoke Object Authority (RVKOBJAUT). Use these CL commands for any object on the system. Use the statements GRANT and REVOKE only for SQL objects: tables, views, and packages. If authorization needs to be changed to other objects, such as programs or collections, use the GRTOBJAUT and RVKOBJAUT commands.

When objects are created on the system, they are given a default authorization. The user ID that creates tables, views, and packages is given all authority. All other user IDs (the public) are given the same authority they have to the collection or library in which the object is created.

Authority to objects referenced by static or dynamic statements within the package are checked at package run time. If the creator of the package does not have authority to the referenced objects, warning messages are returned when the package is created. At execution time, the user executing the package adopts the authority of the creator of the package. If the creator of the package is authorized to a table, but the user running the package is not authorized, the user adopts the authority of the package creator and is allowed to use the table.

For more information on system security see *AS/400 Security - Reference*.

Represent Data

Products supporting DRDA automatically perform any necessary conversions at the application server. For this to happen the application server CCSID value must be a supported value for conversion by the application requester.

On an application server you should be concerned with the CCSID associated with:

- Servicing job in the communication subsystem

  The CCSID of your servicing job must be compatible with the application requester. This CCSID is established by the user profile of the user ID requesting the connection. OS/400 work management support initializes the job CCSID to the CCSID on the user profile. If a CCSID does not exist on the user profile, work management support gets the CCSID (QCCSID) from the system value. The system value QCCSID is initially set to CCSID 65535.

Before initiating a request to DB2 for AS/400, you should sign on and use the Change User Profile (CHGUSRPRF) to assign an acceptable CCSID value to the user profile of the job that will service the DRDA requests.

- SQL collections

  An SQL collection consists of an OS/400 library object, a journal, a journal receiver, and optionally, an IDDU data dictionary if the WITH DATA DICTIONARY clause is specified on the CREATE COLLECTION statement. The physical and logical files used for some of these objects default to the job CCSID at the time of creation. If you query the data dictionary or the catalog from an application requester that does not support the CCSID value of these files, you might see non-displayable or distorted data. Or the application requester might issue a message saying the CCSID value is not supported. To correct this you need to create a new SQL collection with a job CCSID value that is acceptable to the other system.

  The job CCSID can be changed by using the Change Job (CHGJOB) command. Or for subsequent jobs, use the Change User Profile (CHGUSRPRF) command to change the CCSID value of the user profile. In a CL program, use the Retrieve Job Attributes (RTVJOBA) command to get the current job CCSID. Interactively, use the Work with Job (WRKJOB) command and select option 2, Display Job Definition Attributes on the Work with Job display.

- SQL tables and other DB2 for AS/400 files accessed via DRDA

  An SQL table corresponds to an DB2 for AS/400 physical file within a library of the same name as your collection. The columns of a table also correspond to the field definitions of a physical file. The CCSID values for the table or columns of the table might not be compatible with the application requester. To change this value refer to “Represent Data” on page 82, which describes changing database physical files. A major source of CCSID incompatibility in versions of OS/400 prior to Version 3 Release 1 was that many files or SQL tables were tagged with the CCSID 65535 by default. In Version 3 Release 1, and subsequent releases, the CCSIDs of these files are changed automatically to some other more appropriate value.

Chapter 4. Considerations between DB2 for AS/400 and DB2 Universal Database

This section lists some additional considerations that apply to SQL operations between DB2 for AS/400 and DB2 Client Server Version 2 or DB2 Universal Database Version 5. The rest of the discussion relates to DB2 for OS/2, but in most cases similar considerations apply to DB2 Client Server Version 2 and DB2 Universal Database Version 5 on other platforms, as follows:

1. On the AS/400, tables names are qualified by a collection (or library name) and reside in the DB2 for AS/400 database (one database per AS/400). However, on the PC, tables are qualified by a user ID (the creator of the table), and reside in a particular database (with multiple possible databases on a PC with DB2 for OS/2).

   a. This means a query from DB2 for OS/2 (via DB2 Connect) to DB2 for AS/400 would use the user ID of the target side job (on the AS/400) for the (default) collection name, if the name of the queried table was specified without a collection name. Exercise caution, or the table may not be found.

   b. This also means a query from DB2 for AS/400 to DB2 for OS/2 would have an implied table qualifier, if it's not specified in the query (in the form 'qualifier.table-name'). The DB2 for OS/2 table qualifier (specified as a collection or library by the AS/400 application requestor) defaults to the user ID of the user making this query. Again, exercise caution, or your query may not find the table.

   c. You may want to create the DB2 for OS/2 databases and tables with a common user ID. For DB2 for OS/2 there are no physical collections as there are in DB2 for AS/400 but simply a table qualifier, which is the user ID of the creator.

2. For OS/400 DRDA application requesters to work against DB2 for OS/2 DRDA application servers, you may (depending on the release you are using) need some PTFs applied on the AS/400:

OS/400 V2R3 PTF	OS/400 V3R0.5 PTF	OS/400 V3R1 PTF
SF23100	SF23950	SF23270
SF23205	SF23950	SF23277
SF23101	SF23994	SF23271
SF23722	SF23988	SF23721
SF23987	SF23986	SF23985
SF23990	SF23989	SF23960

   For OS/400 V3R1 you may need PTF SF23634, if you will be using DDCS or DB2 Connect. See item 3 and item 7.b.2 below.

   If you will be using the SQL CALL statement to call stored procedures from DB2 for AS/400 to DB2 for OS/2, you need one of the following PTFs:

V3R1	V3R2	V3R6	V3R7
SF36701	SF36699	SF36700	SF33877

The symptom of this error is a "Distributed Relational Database Architecture (DRDA) protocol error" where the "Data descriptor did not match data" at the application requester (SQL code -30000, SQL state 58008).

For OS/400 V3R1 and V3R6 you will need to CALL DB2 for OS/2 procedures with the procedure name in a host variable as mentioned in item 13 (and item 12) below. For OS/400 V3R2 and V3R7 there are PTFs that allow you to embed the procedure name in the SQL statement without putting it into a host variable. These are:

V3R2	V3R7
SF36535	SF35932

3. DB2 Connect (or DDCS) is needed if DB2 for OS/2 will be a client using the DRDA protocol. It is not needed if DB2 for OS/2 will be used only as a server.
4. It is very important to configure DB2 Connect properly:
   a. Make sure you have the most current levels of DB2 for OS/2 and DB2 Connect. Apply any available FIX paks, if you do not.
   b. Follow the installation and configuration instructions given in the manuals.
5. Of course, if you are using APPC, OS/400 communications must also be configured properly, with a controller and device created for the PC, when DB2 for OS/2 is used either as an application requester or as an application server. Furthermore, regardless of the communications protocol used, there needs to be an entry in the RDB directory for each DB2 for OS/2 database an AS/400 will connect to.

   To set up for APPC communications, do the following:

   a. You may manually create the device and controller descriptions. You may also let the system create them for you, if you have a token ring and the line description AUTOCRTCLT parameter says *YES. Use the WRKLIND command to look at the line description, using option 2 Change. Go down to the parameter 'Autocreate controller', and see what your AUTOCRTCLT value is.

      If your system will autocreate controllers, you can initiate the creation of the necessary controller descriptions. From the CM/2 folder on OS/2, Start Communications and run Subsystem Management. From Subsystem Management look at the details to the SNA subsystem. Here you can look at the Logical Links. Open that up and activate the link to the desired system to autocreate the controller there. The device description will automatically be created later.

   b. The device and controller for the PC on the AS/400 needs to be ACTIVE for the network connection to work between systems. You could have the SWTDSC parameter set to *NO in the controller description so ACTIVE controllers will stay ACTIVE. You could also set the ONLINE parameter to *YES, so that on IPL the controller will go active. (The ONLINE parameter in the device description might also need to be set to *YES). Note that to change

parameters on a controller description it must be VARIED OFF and the controller owner (CTLOWN parameter) needs to be set to *USER.

c. To add an entry in the RDB directory for each DB2 for OS/2 database an AS/400 will connect to, use the ADDRDBDIRE command: the RDB name is the DB2 for OS/2 database name and the remote location name is the name of the workstation.

6. The proper CCSID value (normally 37 for US customers) is needed for any tables (physical files) on the AS/400 used by DB2 for OS/2. You can view the CCSID value with DSPFD, and change the CCSID value for physical files with CHGPF. Furthermore, to successfully connect, you may need to change one of the following: the CCSID of the job, the CCSID of the user profile used, or the system CCSID value (QCCSID) if it is the default 65535. Normally the best place to make this change is in the user profile under which the server job will be running.

7. Before using DB2 Connect to interoperate with an AS/400 server, you must create SQL packages on the AS/400 for application programs and for DB2 Connect utilities.

   a. The DB2 PREP command can be used to process an application program source file with embedded SQL. This processing will create a modified source file containing host language calls for the SQL statements and it will, by default, create an SQL package in the database to which you are currently connected.

   b. To bind the DB2 Connect utilities to any AS/400 DB2 server:

      1)
      ```
      CONNECT TO rdbname
      ```

      2)
      ```
      BIND path@DDCS400.LST BLOCKING ALL SQLERROR CONTINUE
               MESSAGES DDCS400.MGS GRANT PUBLIC
      ```

      Replace `path` in `path@DDCS400.LST` above with the default path `C:\SQLLIB\BND\`, or with your local value if you did not install to the default location.

      Note: PTF SF23624 is needed for OS/400 V3R1 to avoid a -901 SQL code from the DB2 for AS/400 database on the third bind file in the list.

      3)
      ```
      CONNECT RESET
      ```

8. For interactive SQL from DB2 for AS/400 to DB2 for OS/2:

   a. Use session attributes of NAMING(*SQL), DATFMT(*ISO), and TIMFMT(*ISO). Other formats besides *ISO work, but not all, and what is used for the date format (DATFMT) must also be used for the time format (TIMFMT).

   b. Note the correspondence between COLLECTIONs on the AS/400, and table qualifier (the creator's user ID) for DB2 for OS/2. See item 1 in this list of considerations for SQL operations.

c. For the very first interactive session, you MUST also specify COMMIT(*CS) for commitment control; and then (1) RELEASE ALL, (2) COMMIT, and (3) CONNECT TO rdbname (where 'rdbname' is replaced with a particular database). You may also want, at this time, to GRANT EXECUTE ON PACKAGE QSQL400.QSQL0200 TO PUBLIC (or to specific users), so that others can use the SQL PKG created on the PC for interactive SQL.

9. For any programs created on an AS/400 accessing a DB2 for OS/2 database, remember to use the following DB2 for OS/2 commands:

   a.

   ```
   GRANT ALL PRIVILEGES ON TABLE table-name TO user
   ```

   b.

   ```
   GRANT EXECUTE ON PACKAGE package-name (usually the AS/400
     program name) TO user
   ```

   Possibly, specify 'PUBLIC' for user.

10. In the development of AS/400 applications accessing DB2 for OS/2 (V.2.1.1 or earlier) a message (SQL5057) was issued in response to the CRTSQLxxx command that says an SQL package had been created on the PC, even if the package had not been created. This is fixed in the latest level of DB2 for OS/2.

    Furthermore, in older versions of DB2 for OS/2, SQL packages would not be created for OS/400 programs that had anything in the text field of its source member description.

11. C language stored procedures in DB2 for OS/2 cannot use argc and argv as parameters (cannot be of type main()). This differs from AS/400 stored procedures which must use use argc and argv. For DB2 for OS/2 stored procedures, see the examples in the \SQLLIB\SAMPLES subdirectory. Look for OUTSRV.SQC and OUTCLI.SQC in the C subdirectory.

12. For stored procedures in DB2 for OS/2, called by an AS/400, use upper case for the procedure name. The AS/400 currently folds procedure names to upper case. However, this means a procedure on the PC, having the same procedure name, but in lower case, will not be found. For stored procedures on an AS/400, remember the procedure names will be upper case.

13. Also without the appropriate PTF for embedded SQL, a CALL statement from an AS/400 to DB2 for OS/2 will work only if you put the procedure name in a host variable (CALL :host-procedure-name(...)). The V3R7 PTF to fix this is SF35932. The V3R2 PTF is SF36535.

14. Stored procedures on the AS/400 cannot include a COMMIT when they are created to run in the same activation group as the calling program (the proper way to create them). However, for DB2 for OS/2, a stored procedure is allowed to include a COMMIT, but the application designer should be aware that there is no knowledge on the part of DB2 for AS/400 that the commit occurred.

Chapter 5. Connecting DB2 for VSE & VM in a DRDA Network

SQL/DS (DB2 for VM) Version 3 Release 5 provides DRDA remote unit of work application server and application requester support for VM systems. SQL/DS (DB2 for VSE) Version 3 Release 5 provides DRDA remote unit of work application server support for VSE systems.

In addition to this, DB2 for VSE & VM Version 5 Release 1 provides DRDA distributed unit of work application server support for both VM and VSE systems. The emphasis on this chapter is mainly on connecting DB2 for VSE & VM systems to unalike remote DRDA systems. For more information on connecting two DB2 for VSE & VM systems, please refer to the following manuals:

- *VM/ESA Connectivity Planning, Administration and Operation*
- *DB2 for VM System Administration*
- *DB2 for VSE System Administration*

DB2 for VM Overview

Each DB2 for VM database manager can manage one or more databases (one at a time) and is typically referred to by the name of the database it manages currently. This relational database name is unique within a set of interconnected SNA networks.

The various DRDA and VM components involved in distributed database processing are described below. These components enable the DB2 for VM database managers to access local relational databases and to communicate with remote DRDA systems in the SNA network.

AVS

APPC/VTAM support (AVS) is a VM component that enables VM applications to access the SNA network. It provides the logical unit (LU) function as defined by SNA. An LU is referred to as a *gateway* in the VM environment. AVS runs in a group control system as a VTAM application. It converts APPC/VM macro calls into APPC/VTAM macro calls and vice versa. APPC/VM uses AVS to route and translate data streams. AVS allows DB2 for VM requests to be routed between the local VM system and remote SNA locations. AVS must be used whenever DB2 for VM applications or databases are communicating with non-DB2 for VM databases or applications.

On the application requester side, a user must be authorized to connect through an AVS gateway before the requests can be sent. On the application server side, the receiving AVS gateway must also be authorized to connect to the DB2 for VM server machine before AVS can pass on the user's requests. The authorization is done by providing the appropriate IUCV directory control statements in the user machine, the database machine, and the sending and receiving

AVS machines. For details on how to do this, see the *VM/ESA Connectivity Planning, Administration, and Operation* manual.

APPC/VM APPC/VM is the VM assembler-level API that provides a subset of the LU 6.2 function set as defined by SNA. In practical terms, it provides the LU 6.2 verbs that enable DB2 for VM applications to connect to and process in local and remote database managers. The LU 6.2 verbs supported by APPC/VM are listed in the manual *VM/ESA CP Programming Services*.

Communications Directory

The Communications Directory is a CMS NAMES file that serves a specific role in the establishment of APPC conversations between a local VM Application Requester and an Application Server. The directory provides the necessary information for routing and establishing an APPC conversation with the target server. This information includes such items as LU name, TPN, security, mode name, user ID, password, and database name.

DB2 for VM uses the COMDIR tag :dbname to resolve the RDB_NAME to its corresponding routing data.

This special file and its communication function are described in the VM/ESA *Connectivity Planning, Administration, and Operation*.

CRR Coordinated Resource Recovery (CRR) is a VM facility that coordinates the commit or backout of updates of protected resources. Distributed application programs, in cooperation with CRR, use protected conversations to ensure distributed transaction resource integrity.

CRR Recovery Server

The CRR Recovery Server is a component of CRR and runs in its own virtual machine. It is responsible for performing sync point logging and resynchronization functions.

GCS Group control system is a VM component that consists of:

- A shared segment that runs in a virtual machine
- A virtual machine supervisor that bands many virtual machines together in a group and supervises their operations
- An interface between the following program products:
  - Virtual Telecommunications Access Method (VTAM)
  - APPC/VTAM Support (AVS)
  - Remote Spooling Communications Subsystem (RSCS)
  - Control Program (CP)

  GCS supervises the execution of VTAM applications such as AVS in a VM environment. Virtual machines running under the supervision of GCS do not use CMS.

Resource adapter

The resource adapter is the portion of DB2 for VM logic that resides in your virtual machine and enables your application to request access to an DB2 for VM server. The DRDA Application Requester function is integrated into the resource adapter.

TSAF

Transparent Services Access Facility is a VM component that provides communications support between interconnected VM systems. Up to eight VM systems can participate in a TSAF collection, which can be considered analogous to a VM local area network (or wide area network). Each participating VM system must have a TSAF virtual machine in operation. Within a TSAF collection, all user IDs and resource IDs are unique.

DB2 for VM uses TSAF to route distributed database requests to other DB2 for VM machines within the TSAF collection. If the local VM system does not have an AVS virtual machine, DB2 for VM uses TSAF to route DRDA requests to a VM system that does have an AVS virtual machine. AVS allows the request to be forwarded to other TSAF collections and non-DB2 for VM systems.

A TSAF collection is viewed as one or more logical units in the SNA network. Resources defined as global within a TSAF collection can be accessed by remote APPC programs residing anywhere in the collection.

Typically, a TSAF collection operates in stand-alone fashion, independent of VTAM and the SNA network. However, it can cooperate with AVS and VTAM to make its global resources accessible by remote APPC programs residing anywhere in the SNA network. This requires that an AVS machine and a VTAM machine are operating on one or more of the TSAF members. TSAF is described in the VM/ESA *Connectivity Planning, Administration, and Operation* manual.

VTAM

Virtual Telecommunications Access Method provides the network communications support for connectivity. DB2 for VM uses VTAM services through AVS to route connections and requests to remote DRDA systems. VTAM is used *only* for remote requests that access the SNA network.

***IDENT**

AVS and TSAF use the transaction program name (TPN) to route requests between VM systems that are connected via TSAF and AVS. The TPN can be an SNA-registered TPN or a valid alphanumeric name. VM refers to the TPN value as a resource ID. For an DB2 for VM server to be accessible to remote DRDA systems, the DB2 for VM server uses the VM IDENTIFY (*IDENT) system service to define itself as the manager of a global resource ID (TPN). After the server is identified as a global resource, TSAF and AVS can route DRDA requests to the DB2 for VM server, if the received TPN matches the resource ID.

Application Requester Communications Flow Example

The following example shows how each component plays a role in establishing communications between a VM Application Requester and a remote DRDA server. Figure 27 shows how the application requester connects to AVS and uses VTAM to access the SNA network. Access to remote resources is not routed through the local DB2 for VM application server.

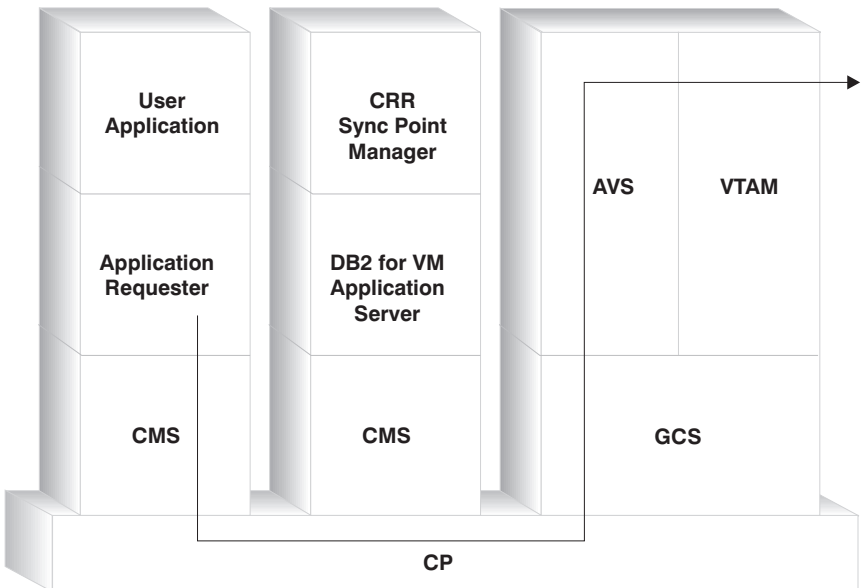

Figure 27. Requesting Access to a Remote Resource

Suppose a DB2 for VM Application Requester that operates in a TSAF collection is to access remote data managed by a DRDA Application Server. By definition, this implies a TSAF machine is operating on the local VM host where the Application Requester resides. Also, an AVS component and a VTAM machine are operating on a VM system in this TSAF collection. AVS and VTAM might also reside on the same system as the Application Requester and the Application Server.

After the VTAM machine starts, it defines the local AVS gateway to the SNA network and activates one or more sessions to use later for establishing conversations.

After the AVS machine starts, it negotiates session limits between the local AVS gateway and the potential partner LUs.

The Application Server might or might not be active. The operator must start it before it can process requests from a like or unlike Application Requester.

The application requester issues an APPC/VM CONNECT statement to establish an LU 6.2 conversation with the Application Server. The CONNECT function uses the CMS

Communications Directory to resolve the relational database name into its associated LU name and TPN that comprise the address of the Application Server in the SNA network. The CMS Communications Directory also determines the level of conversation security and security tokens, such as user ID and password, to pass to the remote site for authorization purposes. If SECURITY=PGM is used, the Application Requester must pass a user ID and password to the Application Server. You can specify the user ID and password in the CMS Communications Directory or in the APPCPASS record defined with the application requester user's CP directory. If SECURITY=SAME is used, then only the VM logon ID of the application requester user is sent to the application server, and no extra password is required.

For example, if you use SECURITY=SAME, the host checks if an AVS machine is operating locally. If it is not, the host establishes a connection between the application requester and the local TSAF machine. The local TSAF machine polls the other TSAF machines in the TSAF collection for the AVS machine and then establishes a connection to it.

The AVS component in the TSAF collection converts the APPC/VM connection request to its APPC/VTAM equivalent function call. AVS then uses an existing session or allocates a new session between its gateway (LU) and the remote LU. AVS then establishes a conversation with the remote LU and passes it the LU name, TPN, security level, and user ID. If the remote LU is also a VM system, the session and conversation are handled by the AVS component running on that system.

Application Server Communications Flow Example

The following example shows how each component plays a role in establishing communications between a remote Application Requester and a local DB2 for VM DRDA server. Figure 28 on page 98 shows that VTAM routes the inbound connection to the specific AVS gateway and then to the Application Server.

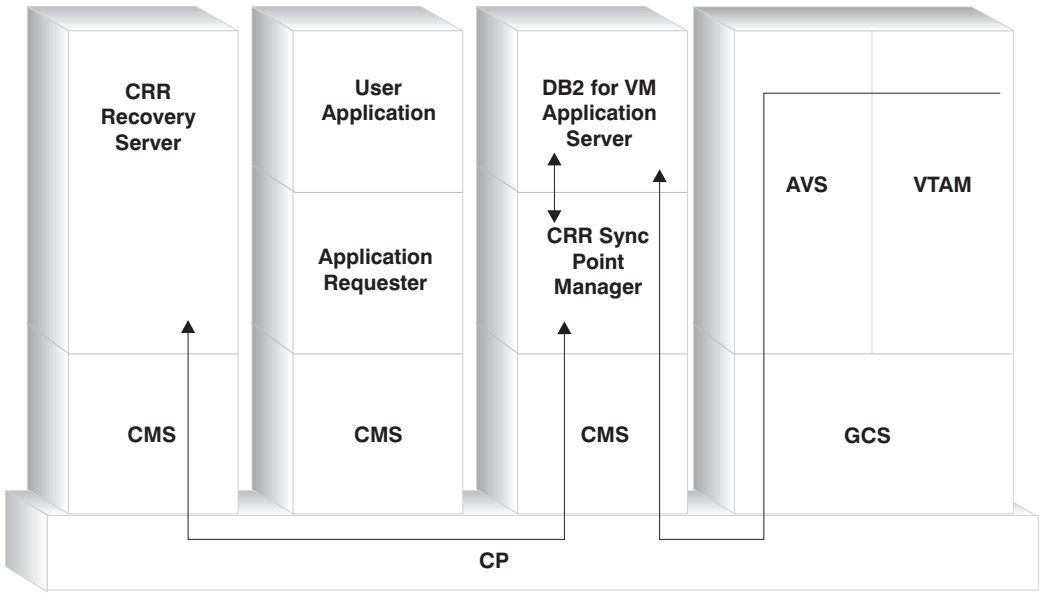

Figure 28. Gaining Access to a Remote Resource

Suppose a DB2 for VM Application Server operates in a TSAF collection. By definition, this implies a TSAF machine is operating on the local VM host where the Application Server resides. Also, an AVS component and a VTAM machine are operating on a VM system in this TSAF collection. AVS and VTAM might also reside on the same system as the Application Requester and the Application Server.

After the VTAM machine starts, it defines the local AVS gateway to the SNA network and activates one or more sessions to use later for establishing conversations.

After the AVS machine starts, it negotiates session limits between the local AVS gateway and the potential partner LUs.

The Application Server might or might not be active. The operator must start it before it can process requests from a like or unlike Application Requester. After the Application Server starts, it uses the *IDENT service to register the resource ID that it manages with the host VM system. Each registration creates an entry in an internal resource table maintained by the VM system.

After the local AVS component establishes the session with its partner LU, it accepts the conversation and passes the TPN, user ID, and password to the VM host for validation. VM searches for the TPN in its internal resource table. This table contains an entry for each resource ID registered through the *IDENT system service. If the TPN search is successful, VM validates the user ID and password with its directory, or RACF or a similar security product. If the validation is successful, AVS establishes a

connection to the Application Server and passes it the user ID for database authorization purposes.

If the table search is unsuccessful, AVS rationalizes that the TPN might reside in another VM system in the TSAF collection and establishes a connection to the local TSAF machine, passing it the user ID, password, and TPN. The TSAF machine polls the other TSAF machines in the TSAF collection. If one of these machines acknowledges the existence of the TPN in its resource table, the local TSAF machine connects to the remote TSAF machine and passes it the user ID and password to be verified with its VM directory. If the validation is successful, the remote TSAF machine connects to the Application Server and passes it the user ID for database authorization.

If the Application Requester wishes to take advantage of DRDA distributed unit of work support, it establishes a protected conversation (i.e. SYNCLEVEL=SYNCPT) with the DB2 for VM Application Server. Before CMS presents the connection to DB2 for VM, it creates a CMS work unit for the protected conversation on the DB2 for VM machine. DB2 for VM then uses this CMS work unit whenever it performs work for the requester. When DB2 for VM begins doing work for the requester, it registers this CMS work unit with the CRR sync point manager. Then, when DB2 receives a "take commit" or "take rollback" indication on the protected conversation, it asks the CRR sync point manager to commit or roll back the unit of work. The CRR sync point manager then drives the commit or rollback, asking the CRR Recovery Server to perform sync point logging when necessary.

Depending on the routing complexity of the connection, the APPC conversation between the Application Requester and Application Server can involve additional systems. However, all the intermediate connections are managed by VM and are transparent to the Application Requester or the user application. The APPC/VM interface lets DB2 for VM Application Servers communicate with APPC application programs located in:

- Same VM system
- Different VM system
- VM system in an SNA network that has AVS and VTAM running
- VM system in a different TSAF collection that has AVS and VTAM running
- Non-VM system in an SNA network that supports the LU 6.2 protocol
- Non-IBM system in an SNA network that supports the LU 6.2 protocol

DB2 for VM Implementation

As shown in Figure 29 on page 100, a VM application must go through the DB2 for VM application requester (resource adapter) to access any DB2 for VM or DRDA Application Server database. A DB2 for VM Application Server database can receive SQL requests from any DB2 for VM or DRDA Application Requester.

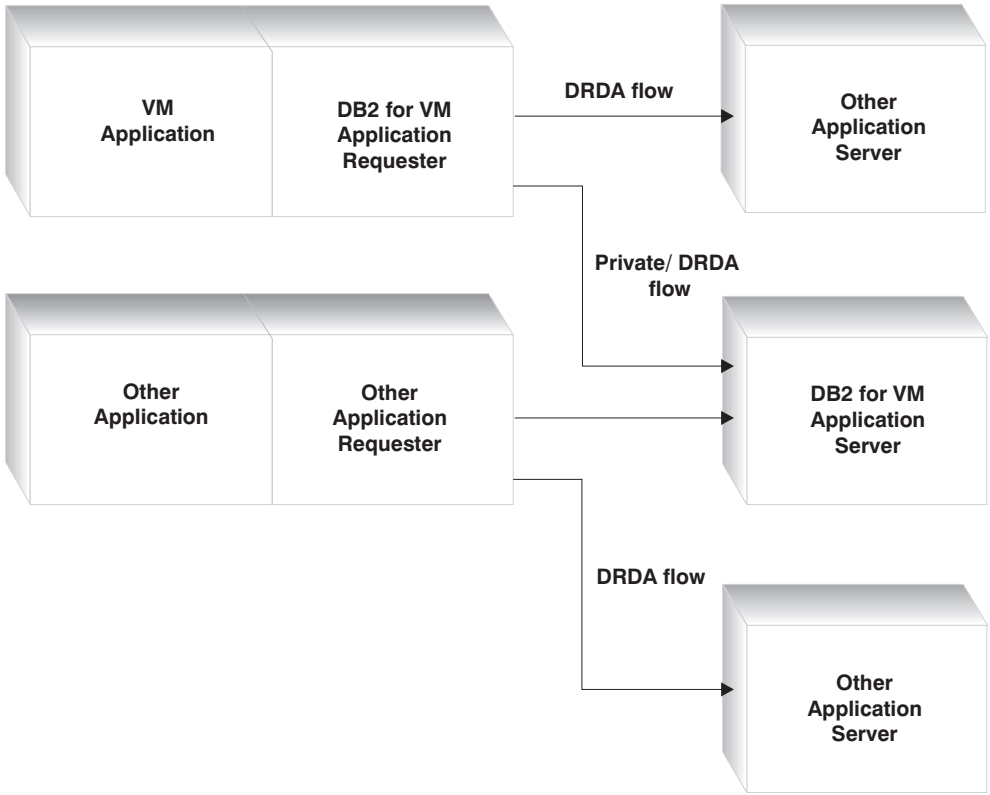

Figure 29. DB2 for VM Application Requester and Application Server

Options for Preprocessing or Running an Application

DB2 for VM supports three processing options on the SQLINIT command that allow the user and the database administrator to enable the distributed database support. The user can specify one of the following SQLINIT options before preprocessing or running the application:

PROTOCOL(SQLDS) Requests the use of the private SQLDS protocol. This is the default option. It can be used between a DB2 for VM application requester and server, in a local or remote environment. The DB2 for VM application server assumes that the requester uses the same CCSIDs as the server. The CCSID defaults[5] set up by the requester via SQLINIT are ignored, and no LU 6.2 LUWID is associated with the conversation. If you use only DB2

5 In DB2 for VM, the application requester and the application server specify the default CCSID by specifying a CHARNAME option for SQLINIT and SQLSTART respectively. The CHARNAME is a symbolic name that is mapped internally to the appropriate CCSIDs.

for VM systems, and the same default CCSID everywhere, then this is the most efficient option.

PROTOCOL(AUTO) Requests the DB2 for VM application requester to find out if the application server is a like or unlike system. It then automatically selects the use of the private SQLDS protocol for a like system, or the DRDA protocol for an unlike system. It can be used between like (local and remote) and unlike systems. If the application server is not set with PROTOCOL=SQLDS, then the application requester and server can have different CCSID defaults. The requests and replies are converted appropriately. AUTO is the recommended option for any of the following cases:

- If you need to access both like and unlike systems
- If the CCSID defaults are different at the requester and server (and the PROTOCOL option of the application server is not SQLDS)
- If you need an LU 6.2 LUWID associated with each conversation so that you can easily trace a task back to its originating site. This is useful if you manage a lot of remote DB2 for VM systems in your distributed database network.

PROTOCOL(DRDA) Forces the DB2 for VM application requester to use only the DRDA protocol to communicate with the application server. You can use this option between like (local and remote) and unlike systems. If the application server is a like system, then DRDA protocol is used between the two DB2 for VM systems. The application requester and application server can have different CCSID defaults. The requests and replies are converted appropriately. You can use this option between two DB2 for VM systems for testing or for specific applications where the use of the DRDA protocol might provide better throughput due to the use of larger buffer size for sending and receiving data.

Table 3 on page 102 compares functional characteristics of the DB2 for VM application requester SQLINIT processing options.

6 Extended dynamic SQL is supported with DRDA flows by converting into static or dynamic statements. Some restrictions apply.

Table 3. Comparison of DB2 for VM Application Requester SQLINIT Processing Options

[SQLDS]	[AUTO]	[DRDA]
Both partners must be DB2 for VM systems	Connects to any DRDA system	Connects to any DRDA system
Can communicate with partner locally, through TSAF or AVS/VTAM	Can communicate with a DB2 for VM system locally, or with a remote DB2 for VM system through TSAF or AVS. With an unlike system, must communicate through AVS.	Can communicate with a DB2 for VM system locally, or with a remote DB2 for VM system through TSAF or AVS. With an unlike system, must communicate through AVS.
Supports static, dynamic, and extended dynamic SQL	Supports static, dynamic, and extended dynamic SQL	Supports static, dynamic, and extended dynamic SQL [6]
CCSIDs defined by SQLINIT for the application requester are ignored by the DB2 for VM application server	CCSIDs defined by SQLINIT for the Application Requester are honored by the DB2 for VM Application Server and proper conversion is performed (if the application server is set to AUTO as well)	CCSIDs defined by SQLINIT for the Application Requester are honored by the DB2 for VM Application Server and proper conversion is performed
Fixed 8K blocksize; OPEN call returns no rows; Application Requester must explicitly close cursor	DB2 for VM to DB2 for VM: SQLDS method; all others: DRDA method	Variable 1K to 32K blocksize; more compact data packaging; OPEN call returns one block of rows; Application Server can implicitly close cursor saving Application Requester from sending a CLOSE call
Can use cursor INSERT and PUTs to insert a block of rows at a time using fixed 8K blocksize	DB2 for VM to DB2 for VM: SQLDS method; all others: DRDA method	PUTs are converted into regular single row inserts and sent out one row at a time
All DB2 for VM-unique commands are supported	DB2 for VM to DB2 for VM: SQLDS method; all others: DRDA method	DB2 for VM operator commands, some DB2 for VM statements, and some ISQL and DBSU commands are not supported (See the *DB2 for VSE & VM SQL Reference*).
LUWID is not supported	LUWID is supported	LUWID is supported

Options for Starting the Database Server Machine

This section describes various options for starting the Database Server Machine.

The PROTOCOL Parameter

The database administrator can specify one of the following options on the `PROTOCOL` parameter when starting the database server machine.

SQLDS The default and recommended option if the application server needs to provide support only for DB2 for VM application requesters or DB2 for VSE application request taking advantage of VSE guest sharing. The application server only uses the private (SQLDS) flow.

The Application Server is sensitive to the processing option selected by the application requester. If a DB2 for VM requester specifies PROTOCOL(SQLDS), the processing on the DB2 for VM server continues normally with private flows. If the DB2 for VM requester specifies PROTOCOL(AUTO), the DB2 for VM server notifies the requester to switch to private flows. No CCSID information is exchanged between the applica-

tion requester and the application server. The application server assumes that the application requester CCSIDs are the same as the application server CCSIDs. If the DB2 for VM requester specifies PROTOCOL(DRDA), the conversation is terminated. If an application requester other than DB2 for VSE & VM attempts to access the DB2 for VM server, the conversation is terminated.

AUTO The recommended option if the application server needs to provide support for both the private protocol and the DRDA protocol. The DB2 for VM application requesters that specify PROTOCOL(SQLDS) or PROTOCOL(AUTO) communicate in the private flow. For an application requester that specifies SQLDS, no CCSID information is exchanged, and the application server assumes that the application requester CCSIDs are the same as the application server CCSIDs. For a requester that specifies AUTO, CCSID information is exchanged, and CCSID conversion of requests and replies are done appropriately. The DRDA flow is required by requesters other than DB2 for VM, or by any DB2 for VM requesters that specify PROTOCOL(DRDA).

The SYNCPNT Parameter

This parameter specifies whether or not a syncpoint manager (SPM) will be used to coordinate DRDA-2 multi-site-read, multi-site-write distributed unit of work activity.

If Y is specified, the server will use a syncpoint manager if possible, to coordinate two-phase commits and resynchronization activity. If N is specified, the application server will not use an SPM to perform two-phase commits. If N is specified, the application server is limited to multi-site-read, single-site-write distributed units of work and it can be the single write site. If Y is specified, but the application server finds that a syncpoint manager is not available, then the server will operate as if N was specified.

The default is SYNCPNT=Y when PROTOCOL=AUTO. When PROTOCOL=SQLDS, the SYNCPNT parameter is set to N.

Setting Up the Application Requester in a VM Environment

DB2 for VM implements the DRDA Application Requester support as an integral part of the resource adapter that resides on the end user virtual machine with the application. You can use the Application Requester support even when the virtual machine of the local database managers is not active. You can activate the DRDA application requester support by running the SQLINIT EXEC with PROTOCOL(AUTO) or PROTOCOL(DRDA) (see “Options for Preprocessing or Running an Application” on page 100).

When DB2 for VM acts as an Application Requester, it can connect to a DB2 for VM Application Server or any other product server that supports the DRDA architecture. For the DB2 for VM Application Requester to provide distributed database access, you need to know how to do the following:

- "Provide Network Information" on page 104. The application requester must be able to accept RDB_NAME values and translate them into SNA NETID.LUNAME values. DB2 for VM uses the CMS Communications Directory to catalog RDB_NAMEs and their corresponding network parameters. The Communications Directory enables the application requester to pass the required SNA information to VTAM when issuing distributed database requests.
- "Provide Security" on page 111. For the application server to accept remote database requests, the Application Requester must provide the security information required by the application server. DB2 for VM uses the Communications Directory and CP directory on the application requester side, and the CP directory or RACF optionally on the application server side to provide required network security information when issuing distributed database requests.
- "Represent Data" on page 115. The application requester must have a CCSID that is compatible with the application server.

Provide Network Information

Much of the processing in a distributed database environment requires messages to be exchanged with other locations in your network. To perform this process correctly, take the following steps:

1. Define the local system
2. Define the remote systems
3. Define the communications subsystem
4. Set RU sizes and pacing
5. Prepare the DB2 for VM Application Requester

Defining the Local System

The DB2 for VM application requester and the DB2 for VM application server are independent of each other. The DB2 for VM application requester directs connection requests directly to local or remote application servers. It does not, however, define itself as the target of inbound connection requests. Only the DB2 for VM application server can accept (or reject) inbound connection requests. Therefore, the DB2 for VM application requester does not identify an RDB_NAME and TPN for itself, as DB2 for OS/390 does.

Define the DB2 for VM application requester to the SNA network as follows:

1. Define AVS gateway names using VTAM APPL definition statements.

   The application requester must have defined gateway names (for example, the LU names) to route its outbound requests into the network. Figure 30 on page 105 shows an example of this. These statements reside on the VTAM virtual machine. When VTAM starts, the gateways are identified to the network but are not activated until the controlling AVS virtual machine starts. Each AVS virtual machine can define multiple gateways on a VM host.

```
         VBUILD TYPE=APPL
************************************************************
*                                                          *
*     Gateway Definition for Toronto DB2 for VM System     *
*                                                          *
************************************************************
TORGATE  APPL  APPC=YES,                                   X
               AUTHEXIT=YES,                               X
               AUTOSES=1,                                  X
               DMINWNL=10,                                 X
               DMINWNR=10,                                 X
               DSESLIM=20,                                 X
               EAS=9999,                                   X
               MAXPVT=100K,                                X
               MODETAB=RDBMODES,                           X
               PARSESS=YES,                                X
               SECACPT=ALREADYV,                           X
               SYNCLVL=SYNCPT,                             X
               VPACING=2
```

Figure 30. Example of an AVS Gateway Definition

The following list describes the VTAM APPL statement keywords that are applicable to topics in this manual. (The VTAM APPL statement supports many more keywords than are shown here).

TORGATE VTAM uses the APPL statement label as the gateway (LU) name. In Figure 30, the gateway TORGATE is defined. The VTAM APPL statement does not specify the NETID. The NETID is automatically assigned for all VTAM applications in the VTAM system.

AUTOSES=1 Gateway TORGATE specifies that one SNA contention winner session automatically starts when you issue an APPC Change Number of Sessions (CNOS) command. You must supply a nonzero value with AUTOSES for AVS to be informed in all cases when CNOS processing fails. You do not have to automatically start all the APPC sessions between any two distributed database partners. If the AUTOSES value is less than the contention winner limit (DMINWNL), VTAM delays starting the remaining sessions until they are required by a distributed database application.

DMINWNL=10 Gateway TORGATE specifies that this DB2 for VM system is the contention winner on at least 10 sessions. CNOS processing uses the DMINWNL parameter for the default, but it can be overridden for any given partner by issuing the AGW CNOS command from the AVS virtual machine.

DMINWNR=10 Gateway TORGATE specifies that this partner system is the contention winner on at least 10 sessions. CNOS processing uses the DMINWNR parameter for the default, but it can be overridden for any given partner by issuing the AGW CNOS command from the AVS virtual machine.

DSESLIM=20 The total number of sessions (both winner and loser) allowed between gateway TORGATE and all partner distributed systems for a specific mode group name is 20. CNOS processing uses the DSESLIM parameter as the default, but it can be overridden for any given partner by issuing the AGW CNOS command from the AVS virtual machine. If the partner cannot support the number of sessions specified by the DSESLIM, DMINWNL, or DMINWNR parameters, the CNOS process negotiates new values for these parameters that are acceptable to the partner.

EAS=9999 An estimate of the total number of sessions that are required by this VTAM LU.

MODETAB=RDBMODES
The name of the VTAM mode table is RDBMODES. This table contains all the mode names this gateway can use to communicate with other distributed database partners.

SECACPT=ALREADYV
This is the security acceptance parameter that identifies the highest APPC conversation security level this gateway supports when it is presented with a distributed database request from a remote partner. SECACPT=ALREADYV is recommended The ALREADYV option supports the following security levels:

SECURITY=NONE, a request containing no security information. DB2 for VM rejects DRDA requests using this security level.
SECURITY=PGM, a request containing the requester's user ID and password. DB2 for VM accepts DRDA requests using this security level.
SECURITY=SAME indicates an already-verified request that contains only the requester's user ID.

SYNCLVL=SYNCPT
The SYNCLVL parameter specifies the synchronization support level for AVS. A value of SYNCPT indicates that synchronization levels of NONE, CONFIRM and SYNCPT are supported. If this AVS gateway is to be used for DRDA-2 distributed unit of work activity at a DB2 for VM server, specify a value of SYNCPT. If distributed unit of work activity will NOT be done, then specify a value of CONFIRM (indicating that NONE and CONFIRM are supported but not SYNCPT).

VERIFY=NONE Identifies the level of SNA session security (partner LU verification) required by this DB2 for VM system. The NONE value indicates that partner LU verification is not required.

DB2 for VM does not restrict your choice for the VERIFY keyword, but the VTAM version you are running can influence this choice. In a nontrusted network, DB2 for VM recommends coding VERIFY=REQUIRED. If you choose VERIFY=OPTIONAL, VTAM performs partner LU verification only for those partners that provide the support. VERIFY=REQUIRED causes VTAM to reject partners that cannot perform partner LU verification.

VPACING=2 This parameter sets the session pacing count used between the partner LU and this gateway. Session pacing is very important for distributed database systems.

2. Activate the gateway.

   Gateway enabling is performed from the AVS virtual machine operating on the same host (or other hosts within the same TSAF collection) as the DB2 for VM Application Requester. Include an AGW ACTIVATE GATEWAY GLOBAL command in the AVS machine's profile or issue this command interactively from the AVS machine console to automatically enable the gateway each time AVS is started.

3. Use the AGW CNOS command to negotiate the number of sessions between the gateway and each of its partner LUs.

   Ensure that the MAXCONN value in the CP directory of the AVS gateway machine is large enough to support the total number of sessions required.

   Issue the AGW DEACTIVE GATEWAY command from the AVS virtual machine to disable the gateway. The gateway definition remains. The gateway can be enabled again at any time using the AGW ACTIVATE GATEWAY GLOBAL command.

   See the *VM/ESA Connectivity Planning, Administration and Operation* for AVS command formats.

4. Make sure that the VTAM NETID is defined to the DB2 FOR VM DBMS during installation.

   The NETID of the host (or other hosts within the same TSAF collection) where the Application Requester resides is supplied by VTAM as the request enters the network. The NETID is stored in the CMS file SNA NETID and resides in the DB2 for VM production disk accessed by the Application Requester. The Application Requester uses this NETID for the generation of the LUWID that flows with each conversation.

Defining the Remote Systems

You must define the remote systems by registering the LU names that enable VTAM to locate the desired network destination. When AVS starts, it identifies the global gateway names (the LU names) available for routing SQL requests into the network to VTAM. A gateway name must be unique within the set of LU names recognized by the local VTAM system so that both inbound and outbound requests are routed to the proper LU

name. This is the best way to ensure gateway name uniqueness throughout the user network. This in turn simplifies the VTAM resource definition process.

When a DB2 for VM application requests data from a remote system, DB2 for VM searches the CMS Communications Directory for the following information relating to the remote system:

- Gateway name (local LU name)
- Remote LU name
- Remote TPN
- Conversation security level required by the application server
- User ID identifying application requester at the application server
- Password authorizing application requester at the application server
- Mode name describing session characteristics to use to communicate with the Application Server
- RDB_NAME

The CMS Communications Directory is a CMS file with file type NAMES, which is created and managed by a DB2 for VM system administrator. As the administrator, you can use XEDIT to create this file and add the desired entries to identify each potential DRDA partner. Each entry in the directory is a set of tags and their associated values. Figure 31 shows a sample entry. When a search is performed, the search key is compared to the :dbname tag value of each entry in the file until a match is found or the end of the file is reached. In the example in Figure 31, the sales manager in Toronto wants to create a monthly sales report for the Montreal branch by accessing data remotely from the MONTREAL_SALES database.

```
 SCOMDIR  NAMES    A1  V 132  Trunc=132 Size=10 Line=1 Col=1 Alt=8
====>
00001  :nick.MTLSALES
00002                 :tpn.SALES
00003                 :luname.TORGATE MTLGATE
00004                 :modename.BATCH
00005                 :security.PGM
00006                 :userid.SALESMGR
00007                 :password.GREATMTH
00008                 :dbname.MONTREAL_SALES
00009
```

Figure 31. A sample entry in a CMS Communications Directory

The :tpn tag identifies the transaction program name that activates the Application Server. The first part of the :luname tag identifies the AVS gateway (local LU) used to gain access to the SNA network. The second part identifies the remote LU name. The :modename tag identifies the VTAM mode that defines the characteristics of the sessions allocated between the local and remote LUs. Request unit (RU) size, pacing, and class of service (COS) are examples of such characteristics. The :security tag indicates the level of security to use on the conversation connecting the Application Requester to the Application Server.

The CMS Communications Directory is on a public system disk accessible to all application requesters in a particular VM system. Any program or product that requires remote access through VTAM can use the CMS Communications Directory.

You can access two levels of the CMS Communications Directory: system-level and user-level. For example, you can create a system-level directory on a public system disk accessible to all Application Requesters in a particular VM system. You can also create your own user-level directory to override existing entries or introduce new entries not appearing in the system-level directory. The user-level directory is searched first, and if the search fails, then the system-level directory is searched. The system-level directory is an extension of the user-level directory; it is searched only if the values are not found in the user-level directory.

Each of these directories is identified to the application and activated through the CMS SET COMDIR command. For example, you can use the following command sequence to identify both system and user-level directories (on the S and A minidisks respectively) but choose to activate only the system-level directory for searches:

```
SET COMDIR FILE SYSTEM SCOMDIR NAMES S

SET COMDIR FILE USER UCOMDIR NAMES A

SET COMDIR OFF USER
```

The CMS Communications Directory is described in detail in *VM/ESA Connectivity Planning, Administration and Operation*. The CMS SET COMDIR command is described in *VM/ESA CMS Command Reference*.

Defining the Communications Subsystem

In the VM environment, a combination of components performs communication management. The components involved in the communication among unlike DRDA systems are APPC/VM, CMS Communications Directory, TSAF, AVS, and VTAM.

APPC/VM is the LU 6.2 assembler-level API that the DB2 for VM Application Requester uses to request communications services. The CMS Communications Directory provides the routing and security information of the distributed partner system. AVS activates the gateway and translates outbound APPC/VM flows into APPC/VTAM flows, and inbound APPC/VTAM flows into APPC/VM flows.

APPC/VM, TSAF, and AVS rely on the CMS Communications Directory, VTAM, and *IDENT to route requests to the proper DRDA partner.

For VTAM to communicate with the partner applications identified in the CMS Communications Directory, you must provide the following information:

1. Define the LU name of each Application Requester and Application Server to VTAM. The placement and syntax of these definitions is dependent on how the remote system is logically and physically connected to the VTAM system.

2. Create an entry in the VTAM mode table for each mode name specified in the CMS Communications Directory. These entries describe the request unit (RU) size, pacing window size, and class of service for a particular mode name.
3. If you intend to use partner LU verification (session-level security), supply VTAM and RACF profiles (or equivalent) for the verification algorithm.

AVS session limit considerations: When an application requester uses AVS to communicate with a remote application server, a connection is initiated. If this connection causes the established session limit to be exceeded, AVS defers the connection to a pending state until a session becomes available. When a session becomes available, AVS allocates the pended connection on the session, and control is returned to the user application. To avoid this situation, plan for peak usage by increasing session limit to allow for some additional connections. Ensure that the MAXCONN value in the CP directory of the AVS machine is large enough to support peak usage by the APPC/VM connections.

Setting RU Sizes and Pacing

The entries you define in the VTAM mode table specify request unit (RU) sizes and pacing counts. Failure to define these values correctly can have an adverse effect on all VTAM applications.

After choosing request unit (RU) sizes, session limits, and pacing counts, consider the impact these values can have on your existing SNA network. You should review the following items when you install a new distributed database system:

- For VTAM CTC connections, verify that the MAXBFRU parameter is large enough to handle your RU size plus the 29 bytes VTAM adds for the SNA request header and transmission header. MAXBFRU is measured in units of 4K bytes, so MAXBFRU must be at least 2 to accommodate a 4K RU.
- For NCP connections, make sure that MAXDATA is large enough to handle your RU size plus 29 bytes. If you specify a RU size of 4K, MAXDATA must be at least 4125.

  If you specify the NCP MAXBFRU parameter, select a value that can accommodate your RU size plus 29 bytes. For NCP, the MAXBFRU parameter defines the number of VTAM I/O buffers that can hold the PIU. If you choose an IOBUF buffer size of 441, MAXBFRU=10 processes a 4K RU correctly, because 10*441 is greater than 4096+29.
- The *DRDA Connectivity Guide* describes how to assess the impact your distributed database has on the VTAM IOBUF pool. If you use too much of the IOBUF pool resource, VTAM performance is degraded for all VTAM applications.

Prepare the DB2 for VM Application Requester

The DB2 for VM application requester may not have DRDA support installed. Follow the following steps to prepare the DB2 for VM application requester for DRDA communications:

1. Use the ARISDBMA exec to install the DRDA support:

- Use "ARISDBMA DRDA(ARAS=Y)" if installing support for the requester and server.
- Use "ARISDBMA DRDA(AR=Y)" if installing support for the requester only.

See the *DB2 for VM System Administration* manual for details.

2. After issuing ARISDBMA, rebuild the DB2 for VM ARISQLLD LOADLIB. See the chapter *Using a DRDA Environment* of the *DB2 for VM System Administration* manual for more details.

Provide Security

When a remote system performs distributed database processing on behalf of an SQL application, it must be able to satisfy the security requirements of the Application Requester, the Application Server, and the network connecting them. These requirements fall into one or more of the following categories:

- Selection of end user names
- Network security parameters
- Database manager security
- Security enforced by an external security subsystem

Selecting End User Names

In both SQL and LU 6.2, end users are assigned a 1- to 8-character user ID. This user ID value must be unique within a particular operating system, but is not necessarily unique throughout the SNA network. For example, there can be a user named JONES in the TORONTO system and another user named JONES on the MONTREAL system. If these two users are the same person, no conflict exists. However, if the JONES in TORONTO is not the same person as the JONES in MONTREAL, the SNA network (and consequently the distributed database systems within that network) cannot distinguish between JONES in TORONTO and JONES in MONTREAL. If no steps are taken to prevent this situation, JONES in TORONTO can use the privileges granted to JONES in MONTREAL and vice versa.

To eliminate naming conflicts, DB2 for VM provides support for end user name translation. However, the system does not enforce translation of user IDs. If system-enforced translation is required, you should ensure that proper inbound translation is performed at the application server.

Outbound translation is performed using the CMS Communications Directory. An entry in the CMS Communications Directory must specify `:security.PGM`. In this case, the corresponding values in the :userid and :password tags flow to the remote site (Application Server) in the connection request.

By creating the entry shown in Figure 32 on page 112, the user with ID JONES on the local (TORONTO) system is mapped to user ID JONEST when he connects to the MONTREAL_SALES_DB Application Server on the MONTREAL system. In this way, the user ID ambiguity is eliminated.

```
 UCOMDIR  NAMES    A1  V 132  Trunc=132 Size=10 Line=1 Col=1 Alt=8
====>
00001  :nick.MTLSALES
00002                 :tpn.SALES
00003                 :luname.TORLU MTLGATE
00004                 :modename.BATCH
00005                 :security.PGM
00006                 :userid.JONEST
00007                 :password.JONESPW
00008                 :dbname.MONTREAL_SALES_DB
00009
```

Figure 32. Outbound Name Translation

Network Security

Having selected the end user name that represents the application requester at the remote site (Application Server), the application requester must provide the required LU 6.2 network security information. LU 6.2 provides three major network security mechanisms:

- Session-level security, specified using the VERIFY parameter on the VTAM APPL statement.
- Conversation-level security, specified in the CMS Communications Directory.
- Encryption.

Because the application server is responsible for managing the database resources, the application server dictates which network security mechanisms the application requester must provide. You must record the application server's security requirements in the application requester's communications directory by setting the appropriate value in the :security tag.

The SNA conversation-level security options supported by DRDA are:

SECURITY=SAME

This is also known as already-verified security, because only the end user's ID (logon ID) is sent to the remote system. The password is not sent. This level of conversation security is used when `:security.SAME` is specified in the application requester's communications directory for that application server. When this option is used, outbound end user name translation is not performed. The user ID sent to the remote DRDA site is the CMS user's logon ID. The :userid tag in the CMS Communications Directory is ignored for :security.SAME.

SECURITY=PGM

This option causes both the end user's ID and password to be sent to the remote system (Application Server) for validation. This security option is used when `:security.PGM` is specified in the CMS Communications Direc-

tory entry of the application requester. When this option is used, outbound end user name translation is performed.

DB2 for VM does not support password encryption. The password can be specified in the `:password` tag, or it can be stored in the end user's CP directory entry using an APPCPASS directory statement. The APPCPASS statement is recommended if you want to maximize the security of the password. If the password is not specified in the CMS Communications Directory entry, the user's system (VM) directory entry is searched for an APPCPASS statement.

APPCPASS statement: VM provides the APPCPASS statement to maximize the security of the user ID and password used by the Application Requester to connect to an Application Server. The APPCPASS is flexible in that it allows you to store security information in one of the following ways:

- **User ID and password:** In this case the :userid and :password tags in the CMS Communications Directory must be set to blanks.
- **User ID only:** In this case the :userid tag in the CMS Communications Directory must be set to blanks, and the :password tag must be set to the user's password.
- **Password only:** In this case the :password tag in the CMS Communications Directory must be set to blanks, and the :userid tag must be set to the user's ID.

Figure 33 illustrates the case where the user ID is stored in the user's communications directory and the password is stored in the user's VM directory entry. In the communications directory entry, the user ID is set to MTLSOU, but the password is not set. The password is stored in the user's VM directory entry.

```
 UCOMDIR  NAMES    A1  V 132  Trunc=132 Size=8  Line=1 Col=1 Alt=8
====>
00001  :nick.MTLSALES
00002                 :tpn.SALES
00003                 :luname.TORGATE MTLGATE
00004                 :modename.BATCH
00005                 :security.PGM
00006                 :userid.MTLSOU
00007                 :password.
00008                 :dbname.MONTREAL_SALES_DB
00009
```

Figure 33. Example of a communications directory entry without a password

When APPC/VM initiates the connection between the Application Requester and the Application Server using conversation SECURITY=PGM, it reads the :userid and :password tag values and passes them to the Application Server. If one or both of these tags is set to blanks, it searches the user's VM directory entry for the missing information. In this case, you must have an APPCPASS statement in the VM directory entry as follows:

```
APPCPASS TORGATE MTLGATE MTLSOU Q6VBN8XP
```

This statement tells APPC/VM that the user (Application Requester) requesting the connection via the (local) AVS gateway TORGATE, the partner LU named MTLGATE, and the user ID MTLSOU should send the password Q6VBN8XP to the Application Server. The user is known by these two pieces of identification at the Application Server.

Placing the APPCPASS statement in the VM directory is not an end user task. The end user must place a request with the VM systems programmer to do this.

For more information on conversation-level security and the APPCPASS statement refer to *VM/ESA Connectivity Planning, Administration, and Operation*.

Database Manager Security

As part of the overall distributed database security framework in DRDA, the Application Requester can play a role in controlling which end users are allowed to make distributed database requests. In DB2 for VM, the Application Requester can participate in distributed database security in three ways:

Outbound user name translation

You can use outbound user name translation to control access to a particular Application Server, based on the identity of the end user making the request. DB2 for VM attempts to translate the end user's name before sending the request to the remote site. However, the best way is to have the Application Server perform come-from checking and inbound translation, because VM Application Requester users can potentially override the outbound translation with their CMS User Communications Directory.

Application preprocessing

End users preprocess remote applications to a particular Application Server by using the DB2 for VM SQLPREP EXEC or the Database Service Utility (DBSU) RELOAD PACKAGE command. DB2 for VM does not restrict the use of these services. When an end user preprocesses an application, that user owns the resulting package.

Application execution

For the DB2 for VM end user to run a remote application, the end user must have authority at the remote site (Application Server) to run the remote package associated with the particular application. The creator (owner) of the package is automatically authorized to run the package. Other end users can be given authority to run the package with the DB2 for VM GRANT EXECUTE statement. In this way, the owner of a distributed database application can control the use of the application on a user-by-user basis.

Security Subsystem

The external security subsystem on VM systems is provided by either RACF or equivalent products that provide an interface compatible with RACF. The DB2 for VM Application Requester does not interface directly to the external security subsystem. The external security subsystem is not used to provide passwords for conversation-level security. If you choose to use session-level security, the external security subsystem is

called by VTAM to validate the identity of the remote LU name during partner LU verification.

Represent Data

The application requester must have the appropriate default CHARNAME and CCSID values. Choosing the correct values ensures the integrity of character data representation and reduces performance overhead associated with CCSID conversion.

For example, if your DB2 for VM application requester is generated with code page 37 and character set 697(CP/CS 37/697) for US ENGLISH characters, then the application requester should set the default CHARNAME to ENGLISH. This is because CP/CS 37/697 corresponds to the CCSID of 37, which corresponds to the CHARNAME of ENGLISH.

The default CHARNAME of a newly installed or migrated system is INTERNATIONAL and the CCSID is 500. This is probably *not* correct for your installation. To display the values of the current default CCSIDs, use the following command:

```
SQLINIT QUERY
```

The appropriate CCSID value for the application requester might be one that is not supported by conversion tables at the application server. If this is the case, you can establish the connection by doing one of the following:

- Have the application server update its CCSID conversion table to support the conversion between the application requester default CCSID and the application server default CCSID (refer to the application server product manuals for details on how to add CCSID conversion support).
- Change the application requester default CCSID to one that is supported by the application server. This might cause data integrity problems, and you must be aware of the consequences. An example of such a consequence follows:

  An application requester uses a controller defined with CP/CS 37/697. The application server does not support a conversion from CCSID 37, but does support a conversion from CCSID 285 (this is CHARNAME UK-ENGLISH for SQL/DS).

  If the application requester is changed to use a default CHARNAME of UK-ENGLISH (and CCSID of 285) then data integrity will not be maintained. For example, where a British pound sign character (6) is meant by the application server, the application requester displays a dollar sign ($). Other characters might also be different.

  To change the CCSID value of a DB2 for VM application requester, you must specify the CHARNAME parameter of the SQLINIT EXEC. See the *DB2 for VM System Administration* manual for more detailed information.

The appropriate CCSID value for the application server might be one that is not supported by conversion tables at the application requester. If this is the case, you can establish the connection by doing one of the following:

- Update the conversion table used by the application requester to support the conversion between the application server default CCSID and the application requester default CCSID. See *DB2 for VM System Administration* for details on how to update the SYSTEM.SYSSTRINGS system table. This table is used to create the CMS file ARISSTR MACRO, which is used by the application requester for CCSID conversion support.
- Have the application server change its default CCSID. This should be done only if appropriate, taking into account the goals of choosing the application server default CCSID. The application server default CCSID affects all application requesters that connect to it, the operator terminal used with the application server, and the data stored in tables on the application server.

Checklist for Enabling a DB2 for VM DRDA Application Requester

The following checklist summarizes the steps needed to enable a DRDA Application Requester for DRDA communications, starting with the assumption that your VM system is installed with ACF/VTAM as its teleprocessing access method, and that VTAM definitions needed to communicate with the remote systems, such as NCP definitions are completed.

1. Define the local AVS gateway to VTAM
2. Install DRDA support into the DB2 for VM Application Requester using the ARISDBMA exec.
3. Set up a CMS Communications directory and add any necessary APPCPASS statements to the VM directory of the application VM machine. Use the SET COMDIR CMS command to enable the communications directory.
4. Start up VTAM and AVS so that VM applications can communicate remotely through the SNA network.
5. Issue the SQLINIT exec and specify the DBNAME, PROTOCOL and CHARNAME parameters to indicate the default database, the protocol to be used and the CCSIDs to be used.
6. Prepare applications on the remote server.

Setting Up the Application Server in a VM Environment

The application server support in DB2 for VM allows DB2 for VM to act as a server for DRDA application requesters. The application requester connected to an DB2 for VM application server can be one of the following:

- A DB2 for VM requester
- A DB2 for OS/390 requester
- A OS/400 requester
- A DB2 for Windows 95 & Windows NT requester
- Any DB2 family application requester, including DB2 CONNECT, or any other product that supports DRDA Application Requester protocols can connect to a DB2 for VM application server.

For any application requester connected to a DB2 for VM application server, the DB2 for VM application server allows the application requester to access database objects (such as tables) stored locally at the DB2 for VM application server. The application requester must create a package containing the application's SQL statements at the DB2 for VM application server before the connection can be established.

For the DB2 for VM Application Server to process distributed database requests, you must take the following steps:

1. Define the Application Server to the local communication subsystem.
2. Provide the necessary security.
3. Provide for data representation.

Provide Network Information

Defining the Application Server

For the Application Server to receive distributed database requests, define the Application Server to the local communications subsystem and assign a unique RDB_NAME.

Follow these steps to define the application server:

1. Define the DB2 for VM application server to the SNA network. After selecting the gateway name and RDB_NAME for the DB2 for VM Application Server, follow the procedures described in “Provide Network Information” on page 104. The RDB_NAME you choose for DB2 for VM must be supplied to all users (Application Requesters) that might require connection to the DB2 for VM Application Server.

   The NETID is defined to VTAM as a startup parameter, and all distributed requests from the Application Requester are routed to it correctly. The DB2 for VM Application Server does not set the NETID.

   The DB2 for VM Application Server does not determine which gateway to use to route the inbound distributed requests from the Application Requester. The Application Requester always controls this. In the case of an DB2 for VM Application Requester, the CMS Communications Directory specifies it using the :luname and :tpn tags.

   In order for the DB2 for VM Application Server to support distributed unit of work activity, the Application Requester must select an AVS gateway that has been defined to VTAM using the SYNCLVL=SYNCPT parameter. Make sure that the AVS gateway has been defined to support distributed units of work.

2. Create a CRR recovery server used to manage distributed unit of work activity for DB2 for VM Application Servers on this VM system. To do this, perform the steps to post-install load the IBM-supplied servers and file pools described in the *VM/ESA Installation Guide*. This includes defining a CRR server (VMSERVR) and a CRR file pool (VMSYSR). Make sure that when starting the CRR recovery server, an LUNAME is specified that equals the name of an AVS gateway for which SYNCLVL=SYNCPT was specified.

3. Ensure that the CP directory for the application server machine has an IUCV *IDENT statement. This identifies the server as a global resource.
4. Create an entry in the VTAM mode name table for each mode name that an Application Requester requests. These entries describe session characteristics such as RU size, pacing count, and class of service for a particular mode name.
5. Define session limits for the Application Requesters that connect to the DB2 for VM Application Server. The VTAM APPL statement defines default session limits for all partner systems. To establish unique defaults for a particular partner, use the AGW CNOS command from the AVS virtual machine running at the Application Server site. (Session limits are usually requested by the application requester.)

   After choosing RU sizes, session limits, and pacing counts, consider the impact these values have on the VTAM IOBUF pool.

Mapping the server name to a RESID: A resource ID (RESID) is the VM term for transaction program name. In the VM environment, it is commonly defined as an alphanumeric name up to 8 bytes long. You normally define a RESID that is identical to the server name, to keep administration easy. Figure 34 shows a sample RESID names file.

```
 RESID  NAMES    A1  V 132  Trunc=132 Size=4  Line=1 Col=1 Alt=3
====>
00001  :nick.MTLTPN
00002                 :dbname.MONTREAL_SALES_DB
00003                 :resid.SALES
00004
```

Figure 34. Example of a RESID names file

See Figure 33 on page 113 for the Communications Directory entry that defines this dbname and RESID (as the TPN). If the application server name cannot be the same as the RESID, then the DB2 for VM application server uses a RESID NAMES file to provide the mapping. This mapping is needed if you:

- Use a RESID different from the server name
- Use a server name longer than 8 bytes
- Use a RESID with a 4-byte hexadecimal value, such as the default DRDA TPN X'07F6C4C2'

During installation, the default is to use the server name specified on the SQLDBINS EXEC as the RESID. To create a mapping entry in the RESID NAMES file, specify the RESID parameter on SQLDBINS.

When you start up the database using SQLSTART DB(server_name), DB2 for VM looks up the corresponding RESID and informs VM that this is the resource that VM is to control. If an entry is not found in the RESID NAMES file, DB2 for VM assumes the RESID is the same as the server name and tells VM so. For more information, see the *DB2 for VM System Administration* manual.

Preparing and starting the DB2 for VM Application Server

The DB2 for VM application server may not have DRDA support installed. Follow the following steps to prepare the DB2 for VM application server for DRDA communications:

1. Use the ARISDBMA exec to install the DRDA support:
   - Use "ARISDBMA DRDA(ARAS=Y)" if installing support for the requester and server.
   - Use "ARISDBMA DRDA(AS=Y)" if installing support for the server only.

   See the *VM/ESA System Administration* manual for details.
2. After issuing ARISDBMA, rebuild the DB2 for VM ARISQLLD LOADLIB. See the chapter *Using a DRDA Environment* of the manual *DB2 for VM System Administration* for more details.

Provide Security

When an Application Requester routes a distributed database request to the DB2 for VM application server, the following security considerations can apply:

- Inbound end user name translation
- Network security parameters
- Database manager security
- Security enforced by an external security subsystem

End User Names

In both SQL and LU 6.2, end users are assigned a 1- to 8-byte user ID. This user ID must be unique within a particular operating system, but does not need to be unique throughout the SNA network. To eliminate naming conflicts, DB2 for VM can optionally use the user ID translation function provided by AVS, but only under the following conditions:

- The DB2 for VM Application Server must run in a VM/ESA environment.
- The inbound connection request must be routed through an AVS gateway.
- The partner Application Requester must use conversation SECURITY=SAME (also known as *already verified* in SNA terminology).

If a connection is routed to a server through AVS using the SECURITY=SAME option, then AVS user ID translation is required. The AGW ADD USERID command, issued from the AVS machine, must provide security clearance to the connecting users coming from a specific remote LU or AVS gateway. A mapping must exist for all inbound LUs and user IDs that connect using SECURITY=SAME. The command is flexible; you can accept all user IDs from a particular LU or all remote LUs generically. Or you can accept only a specific set of user IDs from a specific LU.

If you use the AGW ADD USERID command to authorize the inbound (already-verified) user IDs at the local AVS machine, no validation is performed by the host. This means that the authorized ID does not necessarily exist on the host, but the connection is accepted anyway.

Two ways to change the current AVS user ID authorization are:

- Stop AVS, using the AGW STOP command. This nullifies the user ID authorization in its entirety.
- Delete the user ID, using the AGW DELETE USERID command.

As an example, the case of identical user IDs in different cities shows how the AVS translation function can resolve a naming conflict. Suppose a user exists with an ID of JONES in the Toronto system, and another user exists with the same ID in the Montreal system. If JONES in Montreal wants to access data in the Toronto system, the following actions at the Toronto system eliminate the naming conflict and prevent JONES in Montreal from using the privileges granted to JONES at the Toronto system:

1. The AVS operator must use the AGW ADD USERID command to translate the ID of the Montreal user to a local user ID. For example, if the operator issues `AGW ADD USERID MTLGATE JONES MONTJON`, the Montreal user is known as MONTJON at the Toronto system. If all other Montreal users are allowed to connect (connecting via remote LU MTLGATE) and are known locally by their remote user IDs, then the operator must issue the command `AGW ADD USERID MTLGATE * =`. These AVS commands can also be added to the AVS profile so that they are executed automatically when AVS is started.
2. The DBA must use the DB2 FOR VM GRANT command to grant a set of privileges specifically for the translated user ID, MONTJON in this particular case.

These actions can also be performed at the Montreal system to ensure JONES in Toronto does not use privileges granted to JONES in Montreal when accessing remote data at the Montreal system.

The AVS commands that support user ID translation are described in *VM/ESA Connectivity Planning, Administration and Operation*.

Network Security

LU 6.2 provides three major network security features:

- Session-level security
- Conversation-level security
- Encryption

See “Network Security” on page 112 for the discussion on how to specify session-level security for DB2 for VM. The DB2 for VM Application Server uses session-level security the same way the DB2 for VM Application Requester does.

The Application Requester can send either an already-verified user ID (SECURITY=SAME) or a user ID and password (SECURITY=PGM). If a user ID and password are sent, CP, RACF, or an equivalent validates them with the VM directory at the Application Server host. If validation fails, the connection request is rejected; otherwise it is accepted. If the request contains only a user ID, DB2 for VM accepts the request without validating the user ID.

Note: DB2 for VM does not provide encryption capability because VM/ESA does not support encryption.

Database Manager Security

The DB2 for VM Application Server verifies if the user ID given by VM has CONNECT authority to access the database, and then rejects the connection if it does not have authority.

As the owner of database resources, the DB2 for VM Application Server controls the database security functions for SQL objects residing at the DB2 for VM Application Server. Access to objects managed by DB2 for VM is controlled through a set of privileges, which are granted to users by the DB2 for VM system administrator or the owner of the particular object. The DB2 for VM Application Server controls two classes of objects:

- **Packages:** Individual end users are authorized to create, replace, and run packages with the DB2 for VM GRANT statement. When an end user creates a package, that user is automatically authorized to run or replace a package. Other end users must be specifically authorized to run a package at the DB2 for VM Application Server with the GRANT EXECUTE statement. The RUN privilege can be granted to individual end users or to PUBLIC, which allows all end users to run the package.

  When an application is preprocessed on DB2 for VM, the package contains the SQL statements contained in the application program. These SQL statements are classified as:

  - **Static SQL:** This means the SQL statement and the SQL objects referenced by the statement are known at the time the application is preprocessed. The creator of the package must have authority to execute each of the static SQL statements in the package.

    When an end user is granted the privilege to execute a package, the end user automatically has the authority to execute each of the static SQL statements contained in the package. Thus, end users do not need any DB2 for VM table privileges if the package contains only static SQL statements.

  - **Dynamic SQL:** Describes an SQL statement that is not known until the package is run. The SQL statement is built by the program and dynamically preprocessed to DB2 for VM with the SQL PREPARE statement or the EXECUTE IMMEDIATE statement. When an end user runs a dynamic SQL statement, the user must have the table privileges required to execute the SQL statement. Because the SQL statement is not known when the package is created, the end user is not automatically given the required authority by the package owner.

- **SQL objects:** These can be tables, views, and synonyms. DB2 for VM users can be granted various levels of authority to create, delete, change, or read individual SQL objects. This authority is required to preprocess static SQL statements or execute dynamic SQL statements.

Security Subsystem

The use of this subsystem by the DB2 for VM application server is optional. If the application server needs to check the identity of the application requester LU name, VTAM calls the security subsystem to perform the partner LU verification exchange. The decision to perform partner LU verification is made depending on the value specified in the VERIFY parameter of the VTAM APPL statement for the gateway that the DB2 for VM application server uses to receive inbound distributed database requests.

The security subsystem can also be called by CP to validate the user ID and password sent from the application requester. If the security subsystem is RACF and you do not have a RACF system profile, the validation is performed by RACF. If you do have a RACF system profile, for example, RACFPROF, use the following instructions to request this validation from RACF:

```
RALTER VMXEVENT RACFPROF DELMEM (APPCPWVL/NOCTL

RALTER VMXEVENT RACFPROF ADDMEM (APPCPWVL/CTL

SETEVENT REFRESH RACFPROF
```

Represent Data

You must choose the most appropriate default CHARNAME and CCSID for your installation. Using the most appropriate values ensures the integrity of the character data representation and reduces the performance overhead associated with CCSID conversion.

For example, if your DB2 for VM application server is accessed only by local users whose terminal controllers are generated with code page 37 and character set 697 (CP/CS 37/697) for the US ENGLISH characters, then you should set the application server default CHARNAME to ENGLISH. This is because CP/CS 37/697 corresponds to the CCSID of 37, which corresponds to the CHARNAME of ENGLISH.

To eliminate unnecessary CCSID conversion, choose an application server default CCSID to be the same as the CCSID of the application requesters that access your application server most often.

Following is an example of how these two goals can be in conflict:

An application server has less than five application requesters that are local (for VM application requesters, the protocol parameter would be set to SQL/DS) and many (around 100) application requesters that access the application server using the DRDA protocol. The local application requesters have controllers that are defined with CP/CS 37/697. The remote application requesters use CCSID 285.

If the application server default CHARNAME is set to ENGLISH, this keeps the data integrity for the local application requesters, but incurs CCSID conversion overhead for all the remote application requesters.

If the application server default CHARNAME is set to UK-ENGLISH, this avoids the CCSID conversion overhead incurred for all the remote application requesters, but causes data integrity problems for the local application requesters—certain charac-

ters are not displayed correctly at the local application requesters; for example, a British pound sign is displayed as a dollar sign.

To display the current CCSID of the system, query the SYSTEM.SYSOPTIONS table. The application server default CCSID is usually the value of CCSIDMIXED. If this value is zero, then the system default CCSID is the value of CCSIDSBCS. The CHARNAME, CCSIDSBCS, CCSIDMIXED, and CCSIDGRAPHIC values in this table are updated to the values used as the system defaults every time the database is started. The values in this table might not always be the system defaults. A user with DBA authority might have changed these values, though this is not recommended. To change the application server default CCSID, you must specify the CHARNAME parameter of the SQLSTART EXEC the next time the application server is started. See the *VM/ESA System Administration* manual for more detailed information.

For a newly installed database, the application server default CHARNAME is INTERNATIONAL, and the application server default CCSID is 500. This is probably *not* correct for your system. The default CHARNAME for a migrated system is ENGLISH, and the default CCSID is 37.

Checklist for Enabling a DB2 for VM DRDA Application Server

The following checklist summarizes the steps needed to enable a DRDA Application Server for DRDA communications, starting with the assumption that your VM system is installed with ACF/VTAM as its teleprocessing access method, and that VTAM definitions needed to communicate with the remote systems, such as NCP definitions are completed.

1. Define the local AVS gateway to VTAM.
2. Create a CRR Recovery Server. Make sure that the LUNAME specified by the CRR Recovery Server matches the name of an AVS gateway that can handle SYNCLVL=SYNCPNT conversations.
3. Install DRDA support into the DB2 for VM Application Server using the ARISDBMA exec.
4. Add an IUCV *IDENT statement to the CP directory of the VM server machine so it can identify itself as a global resource.
5. Define local user IDs and passwords to CP that will be used by remote Application Requesters. If necessary, map any remote user IDs to local VM user IDs using the AVS AGW ADD USERID command.
6. Create an entry in the VTAM mode name table for each mode that an Application Requester requests.
7. Start up VTAM and AVS so that VM applications can communicate remotely through the SNA network.
8. Establish session limits for all partner systems where the Application Requesters are located.

9. Start up the DB2 for VM application server with the DBNAME, PROTOCOL and SYNCPNT parameters. When the database manager has started up, make sure that it has identified itself as a GLOBAL resource.
10. Prepare applications on the DB2 for VM Application Server.

DB2 for VSE Overview

In the VSE/ESA operating environment, DB2 for VSE provides the application server function in a DRDA environment. The application requester function is not provided. The various DB2 for VSE and VSE components involved in distributed database processing are described in this section. These components enable the DB2 for VSE database management system to communicate with remote DRDA application requesters in an SNA network.

CICS(ISC) The Customer Information Control System (CICS) intersystem communication component provides the SNA LU 6.2 (APPC) functions to the DB2 for VSE application server.

CICS(SPM) The CICS syncpoint management component is integral to DB2 for VSE DRDA distributed unit of work support. It acts as a syncpoint participant and is responsible for coordinating two-phase commit activity at a VSE/ESA system.

CICS(TRUE) The CICS task-related user exit is an interface used by the AXE transaction to interface with the CICS syncpoint manager.

ACF/VTAM CICS(ISC) uses VTAM for VSE to establish, or bind, LU-to-LU sessions with remote systems. DB2 for VSE uses LU 6.2 basic conversations over these sessions to communicate with remote DRDA application requesters.

AXE The APPC-XPCC-Exchange transaction is a CICS transaction activated by the remote DRDA application requester. It routes the DRDA data stream between the remote application requester and the DB2 for VSE application server using the CICS LU 6.2 support and the VSE XPCC functions.

DBNAME Directory
The DBNAME (database name) directory maps an incoming request for conversation allocation to a predetermined application server identified by the incoming TPN. See the *SQL/DS System Administration Guide for VSE* for more details.

XPCC Cross Partition Communication Control is the VSE macro interface that provides data transfer between VSE partitions.

Application Server Communications Flow Example

Figure 35 on page 125 shows how each component plays a role in establishing communications between the DB2 for VSE Application Server and the remote Application Requester.

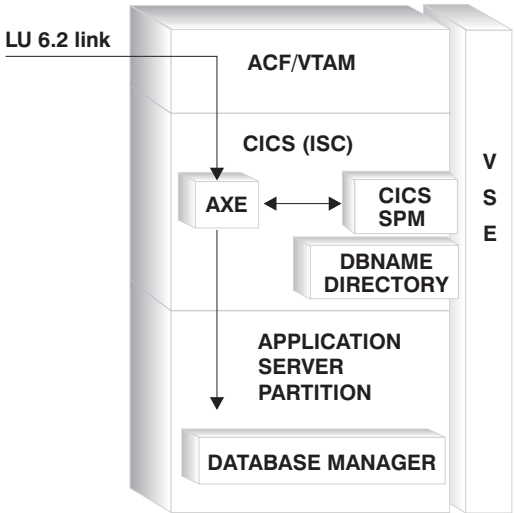

Figure 35. Gaining Access to the Application Server

The application requester issues an APPC ALLOCATE verb with a specific LU name and transaction program name (TPN) to establish an LU 6.2 conversation with the application server. The LU name is used to route the ALLOCATE request through VTAM to CICS. Upon receiving the ALLOCATE verb, CICS verifies that an AXE transaction is defined with that TPN, and performs a CICS sign-on. If the conversation security level for the CICS connection is VERIFY, both user ID and password are expected from the application requester, and are used in the sign-on. The CICS sign-on table (DFHSNT) must be updated with this user ID and password so that the connection is accepted. If the security level is set to IDENTIFY, only the user ID is required, and CICS entrusts the security check to the remote system. If the security check is successful, CICS starts the AXE transaction to route requests and replies between the application requester and an application server. The TPN used by the application requester must also have an entry defined in the DB2 for VSE DBNAME directory that points to an operating DB2 for VSE server within the VSE system.

If the application requester wishes to take advantage of distributed unit of work support, it specifies a SYNCLVL of SYNCPT on the APPC ALLOCATE verb. When the AXE transaction has started, it queries CICS to determine the SYNCLVL of the conversation. If it is SYNCPT, it does the following:

- If necessary, the AXE transaction enables TRUE support so that it can communicate with the CICS syncpoint manager.
- It registers the logical unit of work with the CICS syncpoint manager.

Limitations

Unlike its VM counterpart, the DB2 for VSE application server accepts DRDA flows from remote application requesters. Private protocols are not supported. As a result, VM application requesters cannot access a VSE server with PROTOCOL=SQLDS.

The DB2 for VSE DRDA server cannot route requests from remote application requesters to a DB2 for VM server using VSE guest sharing. Such requests should be sent directly to the DB2 for VM DRDA server.

Application Server Startup Parameters

The RMTUSERS Parameter

The database administrator can specify the RMTUSERS parameter when starting the application server to set the maximum number of remote application requesters that are allowed to connect to the server. This is similar to the MAXCONN value in the VM directory of the DB2 for VM database server machine. This parameter helps to balance the workload between local and remote processing.

When the RMTUSERS value is greater than the number of available DB2 for VSE agents (defined by NCUSER), some remote users must wait for a DB2 for VSE agent to service their request. Normally a DB2 for VSE agent is reassigned to a waiting user at the end of a logical unit of work (LUW). The DB2 for VSE application server supports privileged access that allows a remote user to keep a DB2 for VSE agent for multiple LUWs until the end of the conversation.

The SYNCPNT Parameter

This parameter specifies whether or not a syncpoint manager (SPM) will be used to coordinate DRDA-2 multi-site-read, multi-site-write distributed unit of work activity.

If Y is specified, the server will use a syncpoint manager, if possible, to coordinate two-phase commits and resynchronization activity. If N is specified, the application server will not use an SPM to perform two-phase commits. If N is specified, the application server limited to multi-site-read, single-site-write distributed units of work and it can be the single write site. If Y is specified, but the application server finds that a sync point manager is not available, then the server will operate as if N was specified.

The default is SYNCPNT=Y when RMTUSERS is greater than zero. When RMTUSERS=0, the SYNCPNT parameter is set to N.

Setting Up the Application Server in a VSE Environment

The application server support for DB2 for VSE allows DB2 for VSE to act as a server for DRDA application requesters. The application requester connected to a DB2 for VSE application server can be one of the following:

- A DB2 for VM requester
- A DB2 for OS/390 requester

- A DB2 requester
- A OS/400 requester
- Any DB2 family application requester, including DB2 CONNECT, or any other product that supports DRDA Application Requester protocols can connect to a DB2 for VSE application server.

Provide Network Information

The following steps are required to establish the network connection to the VSE application server:

1. Establish CICS LU 6.2 sessions to the remote systems
2. Define the Application Server

Establishing CICS LU 6.2 sessions

The DB2 for VSE application server communicates with its application requester via CICS LU 6.2 links. The CICS partition used for this purpose must have LU 6.2 links to the remote systems with the application requesters. The *CICS/VSE Intercommunications Guide* contains details on defining and establishing CICS LU 6.2 links with remote systems.

CICS installation and resource definition for LU 6.2 communications

1. Install modules required for ISC.

   You must include the following modules in your system by using SIT or initialization overrides:
   - The EXEC interface programs (specify EXEC=YES or allow it to default).
   - The intersystem communication programs (specify ISC=YES).
   - The terminal control program generated by DFHSG PROGRAM=TCP. A version specifying ACCMETH=VTAM, CHNASSY=YES, and VTAMDEV=LUTYPE6 is required.
2. Install CICS Restart Resynchronization Support

   If the CICS Restart Resynchronization Support was not enabled when the CICS system was installed, you have to update the following CICS tables to enable the CICS Restart Resynchronization capability:

```
DFHJCT    Journal Control Table

          A journal used for the CICS system log must be
          defined in the DFHJCT specifying JFILEID=SYSTEM in a
          DFHJCT TYPE=ENTRY macro.

DFHPCT    Program Control Table

          To generate the DFHPCT entry to use the CICS Restart
          Resynchronization capability, enter:

              DFHPCT TYPE=GROUP,FN=RMI

DFHPPT    Processing Program Table

          To generate the DFHPPT entry to use the CICS Restart
          Resynchronization capability, enter:

              DFHPPT TYPE=GROUP,FN=RMI

DFHSIT    System Initialization Table.

          The DFHSIT macro must include the JCT parameter.
          Specify JCT=YES or JCT=(jj<,....>)  where jj is the
          SUFFIX parameter value specified in th DFHJCT
          TYPE=INITIAL macro defining the CICS system log
          journal data set.
```

3. Define CICS to VTAM for VSE .

   To support LU 6.2 connections, CICS must be defined to VTAM for VSE as a VTAM application major node. The application major node name coded in the VTAM APPL statement is the APPLID for the CICS partition specified in the SIT by the APPLID parameter. It is the LU name used by VTAM (and hence used by the CICS communication partners) to identify the CICS system.

   See Figure 36 on page 129.

```
         VBUILD  TYPE=APPL
*************************************************************************
*                                                                       *
*    LU Definition for Toronto VSE SQL/DS System                        *
*                                                                       *
*************************************************************************
VSEGATE  APPL  ACBNAME=VSEGATE,
               AUTH=(ACQ,SPO,VPACE),
               APPC=NO,
               SONSCIP=YES,
               ESA=30
               MODTAB=RDBMODES,
               PARSESS=YES,
               VPACING=0
```

Figure 36. Example VTAM APPL Definition for CICS

AUTH=(ACQ,SPO,VPACE)
ACQ allows CICS to acquire LU 6.2 sessions.

SPO allows CICS to issue the MODIFY vtamname USERVAR command.

VPACE allows pacing of the intersystem flows.

ESA=30 This option specifies the number of network-addressable units that CICS can establish sessions. The number must include the total number of parallel sessions for this CICS system.

PARSESS=YES Specifies LUTYPE6 parallel session support.

SONSCIP=YES Specifies session outage notification (SON) support. SON enables CICS, in particular cases, to recover a failed session without requiring operator intervention.

APPC=NO This is necessary to let CICS use VTAM macros. CICS does not issue APPCCMD macro instructions.

Note: SYNCLVL=SYNCPT is not required since APPC=NO is specified. CICS manages all SYNCPT syncpoint level activity for distributed units of work.

4. Define links to remote systems using LU 6.2 protocol.
   a. Define all remote LUs to CICS.

      Define all remote LUs using the CEDA DEFINE CONNECTION command on resource definition online online (RDO):

      - Specify the remote LU name on the NETNAME parameter.
      - Specify PROTOCOL=APPC to ensure that LU6.2 protocols are used.
      - Specify AUTOCONNECT=YES and INSERVICE=YES so that the connection, when installed, is put in service automatically and so that sessions are automatically acquired.

- Specify the conversation level security using the ATTACHSEC parameter. ATTACHSEC=IDENTIFY is the minimum security level required by DRDA.
- Specify the session level security using the BINDPASSWORD parameter. The default is no session-level security.

For more details on conversation and session level security, refer to “Provide Security” on page 132.

b. Define groups of LU 6.2 sessions with the remote system.

   For each connection defined above, define groups of parallel sessions for each link to the remote LU using the CEDA DEFINE SESSIONS command:

   - Specify the name of the connection (defined above) on the CONNECTION parameter.
   - Specify the VTAM logmode table entry on the MODENAME parameter.
   - Use the MAXIMUM parameter to specify:
     – The maximum number of sessions
     – The maximum number of sessions that are to be supported as contention winners.

     Specify the values used by the DRDA Application Requester communications software, for example IBM Communications Server for OS/2.

   Note that defining the SENDSize and RECEIVESize with a larger number may improve data transmission rate, however, more virtual storage will also be required across the network. 4 Kilobyte is the size that all layers in the SNA network support. Therefore, when setting up the DRDA Server, set send and receive buffer sizes to 4 Kilobyte. When connections can be made successfully from remote users, adjust these parameters to determine the optimum value.

c. Define user IDs and passwords to CICS

   Define all users in the CICS sign-on table (DFHSNT). You can test the validity of a user ID by performing a CESN logon at a CICS terminal. The local sign-on must be successful.

d. Define the load modules (phases) to CICS using the CEDA DEFINE PROGRAM command:

   1) ARICAXED The AXE transaction
   2) ARICDIRD The DBNAME Directory, and search routine
   3) ARICDAXD DAXP and DAXT transaction handler
   4) ARICDEBD CICS TRUE support enablement handler
   5) ARICDRAD CICS TRUE itself
   6) ARICDR2 DR2DFLT control block

   For each of these, the LANGUAGE=ASSEMBLER option should be specified.

e. For each TPN specified by the application requester, define an AXE transaction using the CEDA DEFINE TRANSACTION command:

- Use the TRANSACTION parameter to specify the TPN
- Specify PROGRAM=ARICAXED to specify the phase
- Use the XTRANID parameter to specify a second hexadecimal transaction name.

At this time, also define the DAXP and DAXT transactions, specifying PROGRAM=ARICDAXD.

Sample definitions:

Refer to *DRDA Connectivity Guide* for sample definitions.

Defining the Application Server

1. Update the DB2 for VSE DBNAME directory.

   Add an entry to the DBNAME directory for each transaction defined above using the CEDA DEFINE TRANSACTION command. With LU 6.2 sessions established, a remote application requester can start a conversation with the DB2 for VSE Application Server. It does so by allocating an LU 6.2 conversation with the Application Server, specifying a TPN (transaction program name). This TPN must be the CICS transaction ID of the AXE transaction responsible for routing requests to or from the DB2 for VSE server. The TPN must be in the DB2 for VSE DBNAME directory mapped to the DB2 for VSE server to be accessed by the application requester. The DB2 for VSE database administrator is responsible for updating the DBNAME directory and informing the remote users of the TPN-to-server mapping.

   Both the TPN and its corresponding server name (database name as defined in the DBNAME directory) must be identified to the application requester:

   - The application requester uses the TPN to initiate the AXE router transaction.
   - The application requester quotes the server name in the initial DRDA flow as the target database name. The DB2 for VSE server uses this server name to verify that the application requester is accessing the right server. A mismatch in server name denies the Application Requester access to the server, and the Application Requester ends the conversation.

2. Use the procedure ARISBDID to build and assemble the DBNAME directory (member ARISDIRD.A).

For more details, please see *DB2 for VSE System Administration* manual.

Preparing and Starting the DB2 for VSE Application Server

1. The AXE transaction maintains an error log that is a CICS temporary storage queue named ARIAXELG. This error log contains useful error messages recording communication problems and abnormal termination of the DRDA sessions. Define this log as “recoverable” using the CICS TST.

2. Run procedure ARIS342D to install the DRDA Application Server support.
3. If necessary, issue the DAXP transaction to specify the default password and language that will be used when CICS TRUE support is enabled for a particular server. See the *DB2 for VSE Operation* manual for more details.
4. Start DB2 for VSE with the DBNAME, RMTUSERS, and SYNCPNT parameter:
   - The DBNAME used must be defined in the DBNAME directory.
   - The RMTUSERS parameter must be nonzero.
   - Specify SYNCPNT=Y to enable distributed unit of work support.
5. All remote users must be authorized by the DB2 for VSE server with different levels of authorization. Refer to *DB2 for VSE Database Administration* for more details.

Problem determination:

- If the application requester succeeded in reaching its partner CICS with a valid TPN (TPN defined in the DBNAME directory), an AXE transaction is started. The use count on program ARICAXED is increased by one (verified by issuing CEMT I PR(ARICAXED)).
- To ensure that a remote user ID is established in the CICS sign-on table, perform a local sign-on using the CESN transaction with the remote user's user ID and password. The local sign-on must be successful.
- When the DB2 for VSE server is running and an application first performs DRDA-2 distributed unit of work activity, TRUE support to a server will enable automatically. Look for message ARI0187I, which indicates that TRUE support was enabled successfully. However if message ARI0190E appears, which indicates that an error occurred while enabling the TRUE, look for prior error messages on the console.
- If your DRDA application program receives sense code X'08063426' or X'FFFE0101', it could be a sign that CICS is running out of sessions. CICS can run out of sessions if all sessions are either in use, or scheduled to be unbound, but the UNBIND has not yet completed. CICS can run out of sessions if there are many concurrent incoming transactions that are short in duration. In this case, increase the number of sessions specified on the CEDA DEFINE SESSIONS MAXIMUM parameter to account for the sessions that are scheduled to be UNBINDed, but the UNBIND has not yet completed.

Provide Security

The DB2 for VSE application server depends on CICS for intersystem communication security. CICS offers several levels of security:

- Bind-time security

  The CICS implementation of the SNA LU 6.2 session-level LU-to-LU verification. The implementation of bind-time security is optional in the LU 6.2 architecture. On the application server side, it can be enabled by supplying a BINDPASSWORD in the CEDA DEFINE CONNECTION command when defining the connection to the

Application Requester. On the application requester, the partner LU that serves the Application Requester must also support bind-time security and use the same password for partner-LU verification.

You can use bind-time security to stop unauthorized remote systems from establishing (binding) sessions with CICS.

- Link security

  Link security can be used to limit a remote system (and its resident DRDA application requester) to attach a certain set of AXE transactions only.

  For example, you can define two AXE transactions: AXE2 with security key 2, and AXE3 with security key 3. Application requesters from a remote system can be assigned an operator security of 3 (for example, using the OPERSECURITY parameter in the CEDA DEFINE SESSION command), allowing them to attach AXE3 only. AXE3 might not have privileged access to the server while AXE2 has privileged access. Refer to *DB2 for VSE System Administration* for a description of privileged access to the application server by remote application requesters.

  Refer to the *CICS Intercommunication Guide* for how to enable link security.

- User security

  The CICS implementation of the SNA LU 6.2 conversation-level security providing end user verification.

  User security validates the user ID with the CICS sign-on table (DFHSNT) before accepting a request to start a conversation. For example, DRDA application requesters not defined in the CICS sign-on table are not allowed to attach an AXE transaction to start a conversation with the DB2 for VSE server. User security level for a remote system can be selected in the CEDA DEFINE CONNECTION command using the ATTACHSEC parameter. The three levels of attach securities are:

  – LOCAL. Not supported by DRDA.

  – IDENTIFY. Equivalent to SECURITY=SAME (or already-verified) in LU 6.2 terminology. With this security level, CICS “trusts” the remote system to verify its users before allowing them to allocate a conversation to the DB2 for VSE server. Only the user ID is required for the CICS sign-on process. However, if the password is also passed, CICS performs the sign-on with the password.

  – VERIFY. Equivalent to SECURITY=PGM in LU 6.2 terminology. With this security level, CICS expects the remote system to send both the user ID and password when allocating the conversation, and rejects the connection if a password is not supplied.

- SNA LU 6.2 session-level mandatory cryptography. Not supported.

Because the application server is responsible for managing the database resources, it dictates which network security mechanisms the application requester must provide. For example, with a DB2 for VM application requester, you must record the application server's conversation-level security requirements in the application requester's commu-

nications directory by setting the appropriate value in the :security tag, as in Figure 37 on page 134:

```
:nick.VSE1     :tpn.TOR3
               :luname.TORGATE VSEGATE
               :modename.IBMRDB
               :security.PGM
               :userid.SALESMGR
               :password.PROFIT
               :dbname.TORONTO3



Where: TOR3    - AXE transaction ID mapped to database TORONTO3.
       TORGATE - VM/APPC gateway.
       VSEGATE - APPLID of the CICS/VSE partition serving as gateway
                 to TORONTO3.
       SALESMGR/PROFIT - USERID/PASSWORD defined in the DFHSNT of
                         VSEGATE, and authorized in TORONTO3
       TORONTO3 - The name specified on the DBNAME startup parameter when
                  the DB2 for VSE application server was started (or the
                  name of the default database determined by the DBNAME
                  Directory if DBNAME was omitted at startup).
```

Figure 37. Sample CMS Communication Directory entry

Database Manager Security

User ID translation is not supported by the VSE application server. CICS uses the user ID transmitted directly from the requester.

After being started by an application requester, the AXE transaction extracts the user ID from CICS and passes it on to the DB2 for VSE server. To set up the required level of user authority on database resources, you must update the user ID into the DB2 for VSE catalog SYSTEM.SYSUSERAUTH.

The DB2 for VSE application server verifies if the user ID given by CICS has CONNECT authority to access the database, and rejects the connection if it does not have authority.

As the owner of database resources, the DB2 for VSE Application Server controls the database security functions for SQL objects residing at the DB2 for VSE Application Server. Access to objects managed by DB2 for VSE is controlled through a set of privileges, which are granted to users by the DB2 for VSE system administrator or the owner of the particular object. The DB2 for VSE Application Server controls two classes of objects:

- **Packages:** Individual end users are authorized to create, replace, and run packages with the DB2 for VSE GRANT statement. When an end user creates a package, that user is automatically authorized to run or replace a package. Other end users must be specifically authorized to run a package at the DB2 for VSE Application Server with the GRANT EXECUTE statement. The RUN privilege can be

granted to individual end users or to PUBLIC, which allows all end users to run the package.

When an application is preprocessed on DB2 for VSE, the package contains the SQL statements contained in the application program. These SQL statements are classified as:

- **Static SQL:** This means the SQL statement and the SQL objects referenced by the statement are known at the time the application is preprocessed. The creator of the package must have authority to execute each of the static SQL statements in the package.

  When an end user is granted the privilege to execute a package, that user automatically has the authority to execute each of the static SQL statements contained in the package. Thus, end users do not need any DB2 for VSE table privileges if the package contains only static SQL statements.
- **Dynamic SQL:** Describes an SQL statement that is not known until the package is run. The SQL statement is built by the program and dynamically preprocessed to DB2 for VSE with the SQL PREPARE statement or the EXECUTE IMMEDIATE statement. When an end user runs a dynamic SQL statement, the user must have the table privileges required to execute the SQL statement. Because the SQL statement is not known when the package is created, the end user is not automatically given the required authority by the package owner.

- **SQL objects:** These can be tables, views, and synonyms. DB2 for VSE users can be granted various levels of authority to create, delete, change, or read individual SQL objects. This authority is required to preprocess static SQL statements or execute dynamic SQL statements.

Represent Data

See “Represent Data” on page 122.

Checklist for Enabling a DB2 for VSE DRDA Application Server

The following checklist summarizes the steps needed to enable a DRDA Application Server, starting with the assumption that your VSE system is installed with ACF/VTAM as its teleprocessing access method, and that VTAM definitions needed to communicate with the remote systems, such as NCP definitions are completed.

1. Install CICS ISC support and Restart Resynchronization support.
2. Define CICS to VTAM for VSE.
3. Assemble the VTAM LOGMODE table with the IBMRDB entry.
4. Assemble the CICS sign-on table with all remote user IDs and passwords defined.
5. Start CICS with the right SIT information:
   - ISC=YES
   - TST=YES, ARIAXELG defined as RECOVERABLE in the DFHTST and assembled
   - APPLID=LU name (as defined in the VTAM APPL statement)
6. Define the remote systems to CICS (RDO can be used):
   - CEDA DEF CONNECTION

- CEDA DEF SESSION
- CEDA DEF PROGRAM
- CEDA DEF TRANSACTION

  These statements should have all definitions under one group, for example, named IBMG. Install the group with: CEDA INSTALL GROUP(IBMG).

7. Update the DBNAME directory (ARISDIRD.A):
   - Define all TPNs listed in the directory to CICS. TPNs not defined to CICS are not usable.
   - Define each DB2 for VSE DRDA application server in the directory with a valid TPN.
8. Run procedure ARISBDID to assemble the updated DBNAME directory.
9. Prepare the DB2 for VSE server:
   - Run procedure ARIS342D to install the DRDA support.
   - If online DB2 for VSE applications (for example, ISQL) are run from the CICS partition, grant schedule authority to the CICS APPLID specified in the CICS SIT table.
   - Grant authority to all remote users.
10. If necessary, run the DAXP CICS transaction.
11. Start DB2 for VSE with the correct RMTUSERS parameter and, optionally, the DBNAME parameter and SYNCPNT parameter.
12. Prepare applications on the VSE DRDA Application Server.

Appendix A. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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.

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

IBM Canada Limited
Department 071
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

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

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM
ADSTAR
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
CICS
C Set++
C/370
DATABASE 2
DatagLANce
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
Distributed Relational Database Architecture
DRDA
Extended Services
FFST
First Failure Support Technology
IBM
IMS
Lan Distance
MVS/ESA
MVS/XA
NetView
OS/400
OS/390
OS/2
PowerPC
QMF
RACF
RISC System/6000
SAA
SP
SQL/DS
SQL/400
S/370
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

Trademarks of Other Companies

The following terms are trademarks or registered trademarks of the companies listed:

C-bus is a trademark of Corollary, Inc.

HP-UX is a trademark of Hewlett-Packard.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

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

Solaris is a trademark of Sun Microsystems, Inc.

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

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

A

B

C

D

E

G

I

L

M

N

O

P

R

S

T

V

W

X

Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the DB2 Universal Database products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

Telephone

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by selecting the "Roadmap to IBM Support" item at: http://www.ibm.com/support/.

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

World Wide Web
http://www.software.ibm.com/data/
http://www.software.ibm.com/data/db2/library/

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

Anonymous FTP Sites
ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

Internet Newsgroups
comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

CompuServe
GO IBMDB2 to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for DB2 Universal Database, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html

Part Number: B2CONNS

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