IBM®

Connectivity Supplement

Version 8

SDB2-CONN-SU

IBM®

Connectivity Supplement

Version 8

SDB2-CONN-SU

Before using this information and the product it supports, be sure to read the general information under *Notices*.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

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

Part 1. Configuring communications manually

Chapter 1. Configuring TCP/IP communications manually

Configuring TCP/IP communications manually between DB2 Connect and a host and iSeries database server

You can manually configure your TCP/IP connection between a DB2 Connect server and a host or iSeries database. TCP/IP is normally configured automatically using the Configuration Assistant (CA).

Prerequisites:

Before you manually configure a TCP/IP connection between DB2 Connect and a host or iSeries database server, ensure that:

- TCP/IP is functional on the DB2 Connect server and host or iSeries system.
- You have identified the following parameter values, using the TCP/IP parameter values worksheet:
  - Hostname (*hostname*) or IP address (*ip_address*)
  - Connection Service name (*svcename*) or Port number/Protocol (*port_number/tcp*)
  - Target database name (*target_dbname*)
  - Local database name (*local_dcsname*)
  - Node name (*node_name*)

Procedure:

To manually configure TCP/IP communications between your DB2 Connect server and a host or iSeries database:

1. Configure TCP/IP on the DB2 Connect server.
2. Catalog the TCP/IP node.
3. Catalog the host or iSeries database as a Database Connection Service (DCS) database.
4. Catalog the host or iSeries database.
5. Bind utilities and applications to the host or iSeries database server.
6. Test the host or iSeries connection.

Note: Due to the characteristics of the TCP/IP protocol, TCP/IP may not be immediately notified of a partner's failure on another host or iSeries. As a result, a client application accessing a remote DB2 server using TCP/IP, or the corresponding agent at the server, may sometimes appear to be hung. DB2 uses the TCP/IP SO_KEEPALIVE socket option to detect when there has been a failure and the TCP/IP connection has been broken.

Related tasks:

- "Configuring TCP/IP on the DB2 Connect server" on page 4
- "Cataloging the TCP/IP node" on page 6
- "Cataloging the database as a Database Connection Service (DCS) database" on page 7

- "Cataloging the database" on page 7
- "Binding utilities and applications to the host or iSeries database server" on page 9
- "Testing the host or iSeries connection" on page 9
- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11

Related reference:

- "TCP/IP parameter value worksheet" on page 129

Configuration tasks

Configuring TCP/IP on the DB2 Connect server

Configuring TCP/IP on the DB2 Connect server is part of the larger task of configuring TCP/IP communications between a DB2 Connect server and a host or iSeries database server.

Procedure:

To configure TCP/IP on the DB2 Connect server:

- Resolve the local host system's IP address.
- Update the services file.

You can now catalog the TCP/IP node.

Related tasks:

- "Resolving local host or iSeries system's IP address" on page 4
- "Updating the services file" on page 5
- "Cataloging the TCP/IP node" on page 6

Configuring TCP/IP tasks

Resolving local host or iSeries system's IP address

Resolving the local host or iSeries system's IP address is part of the larger task of configuring TCP/IP communications between a DB2 Connect server and a host or iSeries database. The DB2 Connect server must know the address of the host or iSeries system to which it is attempting to establish communications.

Note: If your network has a name server, or if you plan to directly specify the IP address (*ip_address*) of the host or iSeries server, you can proceed to cataloging the TCP/IP node.

If a name server does not exist on your network, you may directly specify a hostname that maps to the IP address (*ip_address*) of the host or iSeries system in the local hosts file.

If you plan to support a UNIX client that is using Network Information Services (NIS), and you are not using a domain name server on your network, you must update the hosts file located on your NIS master server.

Table 1. Location of the local hosts and services files

Operating System	Directory
Windows 98	`windows`
Windows NT and Windows 2000	`winnt\system32\drivers\etc`
UNIX	`/etc`

Procedure:

To resolve the local host or iSeries system's IP address, use a text editor to add an entry to the DB2 Connect server's hosts file for the host or iSeries system's hostname.

For example:

```
9.21.15.235     nyx   # host address for nyx
```

where *9.21.15.235* represents the *ip_address*, *nyx* represents the *hostname*, and # represents a comment describing the entry.

If the host or iSeries system is not in the same domain as the DB2 Connect server, you must provide a fully qualified domain name such as *nyx.spifnet.ibm.com*, where *spifnet.ibm.com* represents the domain name.

Your next step is to catalog the TCP/IP node.

Related tasks:

- "Configuring TCP/IP communications manually between DB2 Connect and a host and iSeries database server" on page 3
- "Cataloging the TCP/IP node" on page 6
- "Updating the services file" on page 5

Updating the services file

Updating the services file is part of the larger task of configuring TCP/IP on the DB2 Connect server. Skip this step if you are planning to catalog a TCP/IP node using a port number (*port_number*). You need to update the DB2 Connect server's services file to add the the connection service name and port number of the remote host you want to connect to.

Procedure:

To update the services file, use a text editor to add the connection service name and port number of the remote host to the DB2 Connect server's services file. This file is located in the same directory as the local hosts file.

For example:

```
host1  3700/tcp  # DB2 connection service port
```

where *host1* represents the connection service name, *3700* represents the connection port number, *tcp* represents your communication protocol, and # represents a comment describing the entry.

The port number used on the DB2 Connect server must match the port number used on the host system. Also, ensure that you did not specify a port number that

is being used by any other process. If you are planning on supporting a UNIX client that uses Network Information Services (NIS), you must update the services file located on your NIS master server.

Your next step is to catalog the TCP/IP node.

Related tasks:

- "Cataloging the TCP/IP node" on page 6

Cataloging the TCP/IP node

Cataloging the TCP/IP node is part of the larger task of configuring TCP/IP communications between DB2 Connect and a host or iSeries database server. You must add an entry to the DB2 Connect server's node directory to describe the remote node. This entry specifies the chosen alias (*node_name*), the *hostname* (or *ip_address*), and the *svcename* (or *port_number*) that the client will use to access the remote host.

Prerequisites:

A user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority. You can also log on to the system without these authority levels if you have the `catalog_noauth` option set to `ON`.

Procedure:

To catalog a TCP/IP node:

1. On UNIX, you must set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
2. Catalog the node:

   ```
   catalog tcpip node node_name remote [hostname|ip_address]
      server [svcename|port_number]
   terminate
   ```

   For example, to catalog the remote host *nyx* on the node called *db2node*, using the service name *host1*:

   ```
   catalog tcpip node db2node remote nyx server host1
   terminate
   ```

   To catalog a remote server with the IP address *9.21.15.235* on the node called *db2node*, using the port number *3700*:

   ```
   catalog tcpip node db2node remote 9.21.15.235 server 3700
   terminate
   ```

To change values that were set with the **catalog node** command:

1. Run the **uncatalog node** command in the command line processor as follows:

   ```
   db2 uncatalog node node_name
   ```
2. Recatalog the node with the values that you want to use.

Your next step is to catalog the database as a DCS database.

Related tasks:

- "Configuring TCP/IP on the DB2 Connect server" on page 4
- "Cataloging the database as a Database Connection Service (DCS) database" on page 7

Related reference:

- "CATALOG TCPIP NODE Command" in the *Command Reference*

Cataloging the database as a Database Connection Service (DCS) database

Cataloging the database as a Database Connection Service (DCS) database is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. The remote database must be cataloged as a DCS database so that DB2 Connect can provide access to it.

Prerequisites:

A userID with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

Procedure:

To catalog the remote database as a DCS database:

```
catalog dcs db local_dcsname as target_dbname
terminate
```

where:

- *local_dcsname* represents the local name of the host or iSeries database.
- *target_dbname* represents the host or iSeries database name.

For example, to make *ny* the local database name for DB2 Connect, for the remote host or iSeries database called *newyork*:

```
catalog dcs db ny as newyork
terminate
```

Your next step is to catalog the database.

Related tasks:

- "Cataloging the TCP/IP node" on page 6
- "Cataloging the database" on page 7
- "Cataloging the APPC or APPN node" on page 14

Related reference:

- "CATALOG DCS DATABASE Command" in the *Command Reference*

Cataloging the database

Cataloging the database is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. Before a client application can access a remote database, the database must be cataloged on the host or iSeries system node and on any DB2 Connect server nodes that will connect to it.

When you create a database, it is automatically cataloged on the host or iSeries with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the DB2 Connect server to establish a connection to the remote host or iSeries database.

Prerequisites:

- A userID with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.
- Identify the following parameters:
  - Database name (*database_name*)
  - Database alias (*database_alias*)
  - Node name (*node_name*)

Procedure:

To catalog a database on the DB2 Connect server:

1. On UNIX, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
2. Catalog the database:

   ```
   catalog database database_name as database_alias at
   node node_name authentication auth_value
   ```

   For example, to catalog the DCS known database *ny* so that it has the local database alias *localny*, on the node *db2node*, enter the following commands:

   ```
   catalog database ny as localny at node db2node
     authentication dcs
   terminate
   ```

   To change values that were set with the **catalog** *database* command:

   a. Run the **uncatalog** *database* command in the command line processor as follows:

      ```
      uncatalog database database_alias
      ```

   b. Recatalog the database with the value that you want to use.

Your next step is to bind utilities and applications to the database server.

Related tasks:

- "Cataloging the database as a Database Connection Service (DCS) database" on page 7
- "Binding utilities and applications to the host or iSeries database server" on page 9

Related reference:

- "CATALOG DATABASE Command" in the *Command Reference*

Binding utilities and applications to the host or iSeries database server

Binding utilities and applications to the host or iSeries database server is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. After completing the steps to configure the DB2 Connect server to communicate with the host or iSeries system, you need to bind the utilities and applications to the host or iSeries database server.

Prerequisites:

A userID with BINDADD authority.

Procedure:

To bind the utilities and applications to the host or iSeries database server:

```
connect to dbalias user userid using password
bind bind_path_dir@ddcsmvs.lst blocking all sqlerror continue
     messages mvs.msg grant public
connect reset
```

For example:

```
connect to NYC3 user myuserid using mypassword
bind bind_path_dir@ddcsmvs.lst blocking all sqlerror continue
        messages mvs.msg grant public
connect reset
```

where *bind_path_dir* represents the directory where the .lst files can be found. For example, on Windows the path is usually `\SQLLIB\BND\`.

Your next step is to test the host or iSeries connection.

Related concepts:

- "Binding utilities to the database" in the *Administration Guide: Implementation*

Related tasks:

- "Cataloging the database" on page 7
- "Testing the host or iSeries connection" on page 9

Related reference:

- "BIND Command" in the *Command Reference*

Testing the host or iSeries connection

Testing the host or iSeries connection is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. When you have finished configuring the DB2 Connect server for host or iSeries communications, you need to test the connection on a remote database.

Prerequisites:

- You will need to connect to a remote database to test the connection.
- The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the host or iSeries database server.

Procedure:

To test your host or iSeries connection:

1. Start the database manager by entering the **db2start** command on the host or iSeries database server (if it was not already started).
2. Connect to the remote database:

   ```
   connect to database_alias user userid using password
   ```

   For example, enter the following command:

   ```
   connect to nyc3 user userid using password
   ```

Authentication for connecting to host databases is set while configuring DB2 Connect.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database.

For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command:

```
select tabname from syscat.tables
```

When you are finished using the database connection, enter the **db2 connect reset** command to end the database connection.

Related tasks:

- "Binding utilities and applications to the host or iSeries database server" on page 9

Chapter 2. Configuring APPC communications manually

Configuring APPC communications manually between DB2 Connect and a host and iSeries database server

You can manually configure your APPC connection between a DB2 Connect server and a host or iSeries database. Most APPC communications can be configured automatically using the Configuration Assistant (CA).

Note: You should consider switching to TCP/IP as SNA may no longer be supported in future release of DB2 Connect. SNA requires significant configuration knowledge and the configuration process itself can prove to be error prone. TCP/IP is simple to configure, has lower maintenance costs, and provides superior performance.

Prerequisites:

- APPC is supported on the DB2 Connect server and on the host or iSeries system.
- Identified the parameter values found in the APPC parameter values worksheet.

Restrictions:

The SNA protocol is not supported by DB2 Connect Version 8.1 running on Windows 64–bit platforms (XP 64-bit and .NET Servers 64-bit).

Procedure:

To manually set up a DB2 Connect server to use APPC communications with a host or iSeries database server:

1. Update APPC profiles on the DB2 Connect server.
2. Catalog the APPC or APPN node.
3. Catalog the host or iSeries database as a Database Connection Service (DCS) database.
4. Catalog the host or iSeries database.
5. Bind utilities and applications to the host or iSeries database server.
6. Test the host or iSeries connection.

Related tasks:

- “Updating APPC profiles on the DB2 Connect server” on page 12
- “Cataloging the APPC or APPN node” on page 14
- “Cataloging the database as a Database Connection Service (DCS) database” on page 7
- “Cataloging the database” on page 7
- “Binding utilities and applications to the host or iSeries database server” on page 9
- “Testing the host or iSeries connection” on page 9
- “Configuring TCP/IP communications manually between DB2 Connect and a host and iSeries database server” on page 3

Related reference:

- "APPC parameter value worksheet" on page 130

Configuring tasks

Updating APPC profiles on the DB2 Connect server

Updating APPC profiles on the DB2 Connect server is part of the larger task of configuring APPC communications on the host or iSeries system for DB2 Connect.

Procedure:

To configure DB2 Connect APPC communications to access a remote host or iSeries database server, you need to update the APPC profiles that are appropriate for your network setup:

- Configure an SNA API Client for IBM eNetwork Communications Server for Windows
- Configure Microsoft SNA Serve
- Configure Microsoft SNA Client
- Configure IBM eNetwork Communications Server for AIX
- Configure Bull SNA for AIX
- Configure SNAPlus2 for HP-UX

Your next step is to catalog the APPC or APPN node

Related tasks:

- "Configuring an SNA API Client for IBM eNetwork Communications Server for Windows" on page 12
- "Configuring Microsoft SNA Server" on page 13
- "Configuring Microsoft SNA Client" on page 13
- "Configuring IBM eNetwork Communications Server for AIX" on page 13
- "Configuring Bull SNA for AIX" on page 14
- "Configuring SNAPlus2 for HP-UX" on page 14
- "Cataloging the APPC or APPN node" on page 14

Related reference:

- "APPC communications products configured using the CA" on page 127

Updating APPC profiles subtasks

Configuring an SNA API Client for IBM eNetwork Communications Server for Windows

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8

- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Related tasks:

- "Cataloging the APPC or APPN node" on page 14

Configuring Microsoft SNA Server

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8
- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Related tasks:

- "Configuring Microsoft SNA Client" on page 13
- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11
- "Cataloging the APPC or APPN node" on page 14

Configuring Microsoft SNA Client

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8
- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Related tasks:

- "Configuring Microsoft SNA Server" on page 13
- "Cataloging the APPC or APPN node" on page 14

Configuring IBM eNetwork Communications Server for AIX

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or

iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8

- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Related tasks:

- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11
- "Cataloging the APPC or APPN node" on page 14

Configuring Bull SNA for AIX

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8
- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Configuring SNAPlus2 for HP-UX

The following support has been withdrawn from DB2 Enterprise Server Edition (ESE) for Windows and UNIX Version 8 and DB2 Connect Enterprise Edition (EE) for Windows and UNIX Version 8:

- Two phase commit capability using SNA. Applications that require two phase commit must use TCP/IP connectivity. Two phase commit using TCP/IP to a host or iSeries database server has been available for several releases. Host or iSeries applications which require two phase commit support can use the new capability of TCP/IP two phase commit support within DB2 ESE Version 8
- Applications can no longer access a DB2 UDB ESE server on UNIX or Windows or a DB2 Connect EE server using SNA. Applications can still access host or iSeries database servers using SNA but only using one phase commit.

Related tasks:

- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11
- "Cataloging the APPC or APPN node" on page 14

Cataloging the APPC or APPN node

Cataloging the APPC or APPN node is part of the larger task of configuring APPC communications on the host for DB2 Connect. You must add an entry to the DB2 Connect servers's node directory to describe the remote node.

In most cases, you will add an APPC node entry to the node directory. For Windows 32-bit operating systems, you can alternatively add an APPN node entry if your local SNA node has been set up as an APPN node.

Prerequisites:

A userID with System Administrative (SYSADM) or System Controller (SYSCTRL) authority. You can also log on to the system without these authority levels if you have the `catalog_noauth` option set to `ON`.

Procedure:

To catalog the node:

1. On UNIX, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
2. To catalog an APPC node, specify the chosen alias (*node_name*), Symbolic destination name (*sym_dest_name*), and the APPC security type (*security_type*) that the client will use for the APPC connection. Enter the following commands:

   ```
   catalog "appc node node_name remote sym_dest_name
      security security_type"
   terminate
   ```

   The *sym_dest_name* parameter is case-sensitive and *must* exactly must match the case of the Symbolic destination name you defined previously.

   For example, to catalog a remote database server with the Symbolic destination name *DB2CPIC* on the node called *db2node*, using APPC Security type *program*, enter the following commands:

   ```
   catalog appc node db2node remote DB2CPIC security program
   terminate
   ```
3. To catalog an APPN node, specify the chosen alias (*node_name*), the network ID (**9**), the remote partner LU (**4**), the transaction program name (**17**), the mode (**15**), and the security type. Enter the following commands substituting your own values:

   ```
   catalog "appn node db2node network SPIFNET remote NYM2DB2
            tpname QCNTEDDM mode IBMRDB security PROGRAM"
   terminate
   ```

Your next step is to catalog the database as a Database Connection Service (DCS) database.

Related tasks:

- "Cataloging the database as a Database Connection Service (DCS) database" on page 7

Cataloging the database as a Database Connection Service (DCS) database

Cataloging the database as a Database Connection Service (DCS) database is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. The remote database must be cataloged as a DCS database so that DB2 Connect can provide access to it.

Prerequisites:

A userID with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.

Procedure:

To catalog the remote database as a DCS database:

```
catalog dcs db local_dcsname as target_dbname
terminate
```

where:

- *local_dcsname* represents the local name of the host or iSeries database.
- *target_dbname* represents the host or iSeries database name.

For example, to make *ny* the local database name for DB2 Connect, for the remote host or iSeries database called *newyork*:

```
catalog dcs db ny as newyork
terminate
```

Your next step is to catalog the database.

Related tasks:

- "Cataloging the TCP/IP node" on page 6
- "Cataloging the database" on page 7
- "Cataloging the APPC or APPN node" on page 14

Related reference:

- "CATALOG DCS DATABASE Command" in the *Command Reference*

Cataloging the database

Cataloging the database is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. Before a client application can access a remote database, the database must be cataloged on the host or iSeries system node and on any DB2 Connect server nodes that will connect to it.

When you create a database, it is automatically cataloged on the host or iSeries with the database alias (*database_alias*) the same as the database name (*database_name*). The information in the database directory, along with the information in the node directory, is used on the DB2 Connect server to establish a connection to the remote host or iSeries database.

Prerequisites:

- A userID with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.
- Identify the following parameters:
  - Database name (*database_name*)
  - Database alias (*database_alias*)
  - Node name (*node_name*)

Procedure:

To catalog a database on the DB2 Connect server:

1. On UNIX, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
2. Catalog the database:

   ```
   catalog database database_name as database_alias at
   node node_name authentication auth_value
   ```

   For example, to catalog the DCS known database *ny* so that it has the local database alias *localny*, on the node *db2node*, enter the following commands:

   ```
   catalog database ny as localny at node db2node
     authentication dcs
   terminate
   ```

   To change values that were set with the **catalog** *database* command:

   a. Run the **uncatalog** *database* command in the command line processor as follows:

      ```
      uncatalog database database_alias
      ```

   b. Recatalog the database with the value that you want to use.

Your next step is to bind utilities and applications to the database server.

Related tasks:

- "Cataloging the database as a Database Connection Service (DCS) database" on page 7
- "Binding utilities and applications to the host or iSeries database server" on page 9

Related reference:

- "CATALOG DATABASE Command" in the *Command Reference*

Binding utilities and applications to the host or iSeries database server

Binding utilities and applications to the host or iSeries database server is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. After completing the steps to configure the DB2 Connect server to communicate with the host or iSeries system, you need to bind the utilities and applications to the host or iSeries database server.

Prerequisites:

A userID with BINDADD authority.

Procedure:

To bind the utilities and applications to the host or iSeries database server:

```
connect to dbalias user userid using password
bind bind_path_dir@ddcsmvs.lst blocking all sqlerror continue
     messages mvs.msg grant public
connect reset
```

For example:

```
connect to NYC3 user myuserid using mypassword
bind bind_path_dir@ddcsmvs.lst blocking all sqlerror continue
        messages mvs.msg grant public
connect reset
```

where *bind_path_dir* represents the directory where the .lst files can be found. For example, on Windows the path is usually \SQLLIB\BND\.

Your next step is to test the host or iSeries connection.

Related concepts:

- "Binding utilities to the database" in the *Administration Guide: Implementation*

Related tasks:

- "Cataloging the database" on page 7
- "Testing the host or iSeries connection" on page 9

Related reference:

- "BIND Command" in the *Command Reference*

Testing the host or iSeries connection

Testing the host or iSeries connection is part of the larger task of configuring communications between a DB2 Connect server and a host or iSeries database. When you have finished configuring the DB2 Connect server for host or iSeries communications, you need to test the connection on a remote database.

Prerequisites:

- You will need to connect to a remote database to test the connection.
- The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the host or iSeries database server.

Procedure:

To test your host or iSeries connection:

1. Start the database manager by entering the **db2start** command on the host or iSeries database server (if it was not already started).
2. Connect to the remote database:

   ```
   connect to database_alias user userid using password
   ```

   For example, enter the following command:

   ```
   connect to nyc3 user userid using password
   ```

Authentication for connecting to host databases is set while configuring DB2 Connect.

If the connection is successful, you will get a message showing the name of the database to which you have connected. You are now able to retrieve data from that database.

For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command:

```
select tabname from syscat.tables
```

When you are finished using the database connection, enter the **db2 connect reset** command to end the database connection.

Related tasks:

- "Binding utilities and applications to the host or iSeries database server" on page 9

Part 2. Setting up host or iSeries application requesters

Chapter 3. Setting up OS/390 and z/OS application requesters

Setting up DB2 as an application requester (OS/390 and z/OS)

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

When DB2 for OS/390 and z/OS acts as an application requester, it can connect applications running on the system to remote DB2 Universal Database for OS/390 and z/OS, DB2 UDB for iSeries, and DB2 Server for VSE & VM database servers that implement DRDA application server function.

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 and z/OS uses the DB2 for OS/390 and z/OS Communications Database (CDB) to register RDB_NAMEs and their corresponding network parameters. The CDB allows the DB2 for OS/390 and z/OS application requester to pass the required information to the Communications Server when making distributed database requests over either SNA or TCP/IP connections.

Procedure:

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 DB2 application requester to the local system (SNA) or Define the DB2 application requester to the local system (TCP/IP)
2. Define the remote systems

Related concepts:

- "Data representation (OS/390 and z/OS)" on page 119
- "Security considerations for application requesters (OS/390 and z/OS)" on page 105
- "DB2 for OS/390 and z/OS" on page 69

Related tasks:

- "Defining the DB2 application requester to the local system – SNA (OS/390 and z/OS)" on page 24
- "Defining the DB2 application requester to the local system – TCP/IP (OS/390 and z/OS)" on page 26
- "Defining the remote systems (OS/390 and z/OS)" on page 27
- "Setting up DB2 as an application server (OS/390 and z/OS)" on page 45

Setup tasks

Defining the DB2 application requester to the local system – SNA (OS/390 and z/OS)

Defining the local system is part of the larger task of setting up DB2 for OS/390 and z/OS as an application server. Each program in the SNA network is assigned a NETID and an LU name, so your DB2 for OS/390 and z/OS application requester must have a NETID.LUNAME value (assigned through VTAM) when it connects to the network. Because the DB2 for OS/390 and z/OS application requester is integrated into the local DB2 for OS/390 and z/OS database management system, the application requester must also have an RDB_NAME. In the DB2 for OS/390 and z/OS publications, DB2 for OS/390 and z/OS refers to the RDB_NAME as a *location* name.

Procedure:

To define the DB2 for OS/390 and z/OS application requester to the SNA network:

1. Select an LU name for your DB2 for OS/390 and z/OS system. The NETID for your DB2 for OS/390 and z/OS system is automatically obtained from VTAM when DDF starts.
2. Define the LU name and location name in the DB2 for OS/390 and z/OS *bootstrap data set* (BSDS). (DB2 for OS/390 and z/OS restricts the location name to 16 characters.)
3. Register the selected LU name with VTAM by creating a VTAM APPL definition.
4. Ensure that Extended Security is set to `YES`.

Configuring the DDF BSDS:

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

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

- Use the DDF installation panel DSNTIPR when you first install DB2 for OS/390 and z/OS 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 and z/OS to VTAM. Figure 1 on page 25 shows how to use the installation panel to record location name NEW_YORK3, the LU name NYM2DB2, and password PSWDBD1 in the DB2 for OS/390 and z/OS 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 1. DB2 for OS/390 and z/OS Installation Panel DSNTIPR

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

  Figure 2 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 2. Sample Bootstrap Data Set DDF Definition (for VTAM)

When DDF is started (either automatically at DB2 for OS/390 and z/OS startup or by the DB2 for OS/390 and z/OS START DDF command), it connects to VTAM, passing the LU name and password to VTAM. VTAM recognizes the DB2 for OS/390 and z/OS system by checking the LU name and password (if a VTAM password is required) with the values defined in the DB2 for OS/390 and z/OS VTAM APPL statement. The VTAM password is used to verify that DB2 for OS/390 and z/OS 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 and z/OS.

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.

Register the selected LU name with VTAM by creating a VTAM APPL definition:

After you define the VTAM LU name and password to DB2 for OS/390 and z/OS, you need to register these values with VTAM. VTAM uses the APPL statement to define local LU names. Figure 3 shows a sample definition for the LU name *NYM2DB2*.

```
 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 3. Sample VTAM APPL Definition for DB2 for OS/390 and z/OS

Related tasks:

- "Defining the DB2 application requester to the local system – TCP/IP (OS/390 and z/OS)" on page 26
- "Defining the remote systems (OS/390 and z/OS)" on page 27

Related reference:

- "DB2 Connect VTAM APPL statement keywords" on page 133

Defining the DB2 application requester to the local system – TCP/IP (OS/390 and z/OS)

Procedure:

To define TCP/IP communications with DB2 for OS/390 and z/OS:

1. TCP/IP communications must be enabled on DB2 for OS/390 and z/OS 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 and z/OS 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 and z/OS.
4. Ensure that the TCP/IP already verified security option is set to YES.
5. The DB2 for OS/390 and z/OS BSDS must include additional parameters. Figure 4 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 4. Sample Bootstrap Data Set DDF Definition (for TCP/IP)

Related tasks:

- "Defining the DB2 application requester to the local system – SNA (OS/390 and z/OS)" on page 24
- "Defining the remote systems (OS/390 and z/OS)" on page 27

Defining the remote systems (OS/390 and z/OS)

When a DB2 for OS/390 and z/OS 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 and z/OS system administrator.

Procedure:

As the DB2 for OS/390 and z/OS system administrator, you can use SQL to insert rows in the CDB to describe each potential DRDA partner.

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 Communications Database (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 and z/OS 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 and z/OS 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.

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 Guide* also provides further information about updating the CDB.

Request handling by the Communications Database:

When sending a request, DB2 for OS/390 and z/OS 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 and z/OS installation panel DSNTIPR. To receive TCP/IP requests, you must select a DRDA port and a resynchronization port in DB2 for OS/390 and z/OS 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.

Related tasks:

- "Defining the DB2 application requester to the local system – SNA (OS/390 and z/OS)" on page 24

- "Defining the DB2 application requester to the local system – TCP/IP (OS/390 and z/OS)" on page 26

Chapter 4. Setting up AS/400 application requesters

Setting up DB2 as an application requester – SNA (iSeries)

The iSeries system implements the DRDA application requester (AR) support as an integral part of the OS/400 operating system. Because AR support is part of the OS/400 operating system, it is active whenever the operating system is active.

Procedure:

The AR must be able to accept a relational database name and translate it into network parameters. The iSeries system uses the relational database directory to register relational database names and their corresponding network parameters. This directory allows the iSeries AR 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. When DB2 UDB for iSeries acts as an AR, it can connect to any application server that supports DRDA. For the DB2 UDB for iSeries AR to provide distributed database access:

- Defining the DB2 for iSeries application requester to the local system
- Defining the remote system
- Defining SNA communications

Related concepts:

- "Data representation (iSeries)" on page 119
- "Security considerations for application requesters (iSeries)" on page 111
- "DB2 UDB for iSeries" on page 77
- "Connecting to DB2 UDB using TCP/IP (iSeries)" on page 51

Related tasks:

- "Defining the DB2 application requester to the local system – SNA (iSeries)" on page 31
- "Defining the remote system (iSeries)" on page 32
- "Defining SNA communications (iSeries)" on page 33
- "Setting up DB2 as an application server using SNA (iSeries)" on page 49

Setup tasks

Defining the DB2 application requester to the local system – SNA (iSeries)

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 AR accesses. Any iSeries 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.

Procedure:

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 AR's database is ROCHESTERDB:

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

In the latest versions of OS/400, the local RDB name entry be created automatically if it does not already exist when it is required. The system name in the network attributes will be used as the local RDB name.

Related tasks:

• "Defining the remote system (iSeries)" on page 32

Defining the remote system (iSeries)

Each application server in the distributed database 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 requestor.

Procedure:

To define the remote databases to the local database:

- Add 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 [1] 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 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'.

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

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

Related tasks:

- "Defining SNA communications (iSeries)" on page 33
- "Defining the DB2 application requester to the local system – SNA (iSeries)" on page 31

Defining SNA communications (iSeries)

The iSeries system also allows advanced program-to-program communications (APPC) configurations, which do not provide network routing support. An iSeries distributed database works with either configuration.

AnyNet Support on the iSeries 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.

Procedure:

APPN provides networking support that allows the iSeries system to participate in and control a network of systems without requiring the networking support traditionally provided by a mainframe system. To configure an iSeries 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 iSeries system, if the machine is an end node
   - The network control points, if the iSeries 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 iSeries 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 iSeries system is running to a host system with APPN and as an independent logical unit (LU). The iSeries system automatically creates the device description and attaches it to the appropriate controller description when the session is established. If the iSeries 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 iSeries 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 iSeries estimate for most environments including a distributed database. If the default is taken for RU size, the iSeries system estimates the best value to use. When the iSeries 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.

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.

Related tasks:

- "Defining the DB2 application requester to the local system – SNA (iSeries)" on page 31
- "Defining the remote system (iSeries)" on page 32

Chapter 5. Setting up VM application requesters

Setting up DB2 as an application requester (VM)

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).

Procedure:

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:

- 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.

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 application requester to the local system
2. Define remote systems for the application requester
3. Prepare the application requester or application server for DRDA communications

Related concepts:

- "DB2 for VM" on page 77
- "Security considerations for application requesters (VM)" on page 114

Related tasks:

- "Defining the application requester to the local system (VM)" on page 37
- "Defining remote systems for the application requester (VM)" on page 39
- "Preparing the application requester or application server for DRDA communications (VM)" on page 41
- "Setting up DB2 as an application server (VM)" on page 63

Setup tasks

Defining the application requester to the local system (VM)

Defining the DB2 for VM application requester to the local system is part of the larger task of setting up DB2 for VM as an application requester. 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 and z/OS does.

Procedure:

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 5 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 5. Example of an AVS Gateway Definition

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.

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.

Related tasks:

- "Defining remote systems for the application requester (VM)" on page 39
- "Preparing the application requester or application server for DRDA communications (VM)" on page 41

Defining remote systems for the application requester (VM)

Defining remote systems for the VM application requester is part of the larger task of setting up DB2 for VM as an application requester. 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

Procedure:

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 6 on page 40 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 6 on page 40, 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 6. 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
```

Related tasks:

- "Defining the application requester to the local system (VM)" on page 37
- "Preparing the application requester or application server for DRDA communications (VM)" on page 41

Preparing the application requester or application server for DRDA communications (VM)

Preparing the DB2 for VM application requester or application server is part of the larger task of setting up DB2 for VM as an application requester or as an application server. The DB2 for VM application requester or application server may not have DRDA support installed.

Procedure:

To prepare the DB2 for VM application requester or 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(AR=Y)" if installing support for the requester only.
   - Use "ARISDBMA DRDA(AS=Y)" if installing support for the server only.
2. Rebuild the DB2 for VM ARISQLLD LOADLIB.

For more information, see "Using a DRDA Environment" in the *DB2 Server for VM System Administration* book.

Part 3. Setting up host or iSeries application servers

Chapter 6. Setting up OS/390 and z/OS application servers

Setting up DB2 as an application server (OS/390 and z/OS)

The application server support in DB2 for OS/390 and z/OS allows it to act as a server for DRDA application requesters.

Procedure:

To set up DB2 for OS/390 and z/OS as an application server:

1. Define the application server to the local SNA subsystem.
2. Define the application server to the local TCP/IP subsystem.

Related concepts:

- "Data representation (OS/390 and z/OS)" on page 119
- "DB2 for OS/390 and z/OS" on page 69
- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Related tasks:

- "Defining the application server to the SNA subsystem (OS/390 and z/OS)" on page 45
- "Defining the application server to the local TCP/IP subsystem (OS/390 and z/OS)" on page 47
- "Setting up DB2 as an application requester (OS/390 and z/OS)" on page 23

Setup tasks

Defining the application server to the SNA subsystem (OS/390 and z/OS)

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.

Procedure:

To define the application server to the SNA subsystem:

1. Select the LU name and RDB_NAME to be used by the host DB2 UDB application server. The RDB_NAME you choose for DB2 UDB on the host 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 host DB2 UDB application server with each application requester requiring access, so the application requester can route SNA requests to the host DB2 UDB 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 the host DB2 UDB 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 and z/OS 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).
6. Create entries in the host DB2 UDB CDB to identify which application requesters are allowed to connect to the host DB2 UDB 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 host DB2 UDB 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 7 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 7. 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, the host DB2 UDB 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 the host DB2 UDB to limit access to application requesters and end users identified in the SYSIBM.USERNAMES table. 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 and z/OS to use this row to allow unrestricted access to the host DB2 UDB application server.

In Figure 8, no row contains blanks in the LUNAME column, so the host DB2 UDB 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 8. Identifying Individual Application Requester Connections (SNA)

Related tasks:

- "Defining the application server to the local TCP/IP subsystem (OS/390 and z/OS)" on page 47

Defining the application server to the local TCP/IP subsystem (OS/390 and z/OS)

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 and z/OS Bootstrap Dataset must include the necessary parameters, and you may need to make updates to the DB2 for OS/390 and z/OS Communications Database (CDB).

No CDB updates are required if you will only use inbound database connections, so that if you plan to use DB2 for OS/390 and z/OS 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.

Procedure:

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('        ')
```

For information about setting up TCP/IP at the application server, see *DB2 for OS/390 Installation Guide*.

Related tasks:

- "Defining the application server to the SNA subsystem (OS/390 and z/OS)" on page 45

Chapter 7. Setting up AS/400 application servers (SNA)

Setting up DB2 as an application server using SNA (iSeries)

The application server support on the iSeries system allows it to act as a server for DRDA application requesters. The application requester connected to a DB2 Universal Database (UDB) for iSeries application server can be any client that supports DRDA protocols.

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

Procedure:

To process distributed database requests on the iSeries 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.

The transaction program name used to start an iSeries application server database is the DRDA default X'07F6C4C2'. This transaction program name is defined within the iSeries system to start the application server. The corresponding parameter for TCP/IP connections, when that protocol is supported by DB2 UDB for iSeries, is the port. DB2 UDB for iSeries 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.

Related concepts:

- "Security considerations for application servers (iSeries)" on page 96
- "DB2 UDB for iSeries" on page 77

Related tasks:

- "Configuring TCP/IP on the DB2 Connect server" on page 4
- "Setting up DB2 as an application requester – SNA (iSeries)" on page 31

Chapter 8. Setting up AS/400 application servers (TCP/IP)

Connecting to DB2 UDB using TCP/IP (iSeries)

This topic provides a summary of information contained in *DB2 for AS/400 Distributed Database Programming*, which tells you how to set up DB2® UDB for iSeries:

- As a DRDA® application requester using outbound TCP/IP communications
- As a DRDA application server using inbound TCP/IP communications.

The principles are the same as those described in ″Setting up DB2 UDB for iSeries™ as an application requester using SNA″ and ″Setting up DB2 UDB for iSeries as an application server using SNA″, but the communications configuration steps are much simpler.

Notes:

1. For DRDA communications using TCP/IP the default port number for database connections is 446.
2. The DB2 Universal Database for AS/400 Version 4 Release 2 implementation does not support a two-phase commit (distributed unit of work) over TCP/IP communications.

Summary of DB2 UDB for iSeries information:

The *DB2 for AS/400 Distributed Database Programming* book contains the following sections which you should read and refer to:

- Distributed Relational Database Processing
- DRDA and CDRA Support.
- Configuring a Communications Network using TCP/IP
- DRDA Security using TCP/IP
- Work Management for DRDA Use with TCP/IP
- Setting up the TCP/IP Server
- Managing a TCP/IP Server
- Factors that Affect Blocking for DRDA
- Handling Connection Request Failures for TCP/IP
- Starting a Service Job for a TCP/IP Server
- Cross-Platform Access Using DRDA.

In addition you will need to know:

- TCP/IP port number and hostname information for the server and the requester.
- CCSID and code page information for the server and the requester.
- Userid and password information required when making database connections.

Setup considerations for the DB2 UDB for iSeries DRDA TCP/IP server:

Setting up the DB2 UDB for iSeries DRDA TCP/IP server ensures that the server has been started. The CL command to start the DRDA server (also known as the DDM server) is:

```
STRTCPSVR SERVER(*DDM)
```

The DRDA server can also be started using the Start TCP/IP Server (STRTCPSVR) command without parameters, or with *ALL specified for the SERVER parameter. The DRDA server will be started automatically when TCP/IP is started if this CL command has been issued:

```
CHGDDMTCPA AUTOSTART(*YES)
```

One can verify that the server has been started by issuing the following CL command:

```
WRKUSRJOB USER(QUSER) STATUS(*ACTIVE)
```

This command will show a scrollable list of jobs. If you scroll down a page or so, you should see two lines containing the following information:

```
__   QRWTLSTN    QUSER       BATCH    ACTIVE
__   QRWTSRVR    QUSER       PJ       ACTIVE
```

(There may be repeated occurrences of the QRWTSRVR line, depending on how many prestart server jobs are active.)

The presence of the QRWTLSTN line indicates that the job that listens for DRDA and DDM connections requests is active. This job dispatches work to the QRWTSRVR job(s) as connection requests are received.

Another way to verify that the DRDA server has been started is to issue the STRTCPSVR SERVER(*DDM) command. Look for the 'DDM TCP/IP server already active' message.

The name of the prestart job used for a particular connection can be found by issuing a DSPLOG command such as:

```
DSPLOG PERIOD(('15:55'))
```

where the time specified is earlier than when the connection was made. This will result in a scrollable list of history log entries. Look for an entry like this, which will contain the name of the server job:

```
DDM job 039554/QUSER/QRWTSRVR servicing user SRR on 03/30/01 at 15:57:38.
```

This jobname is useful for looking at the joblog of still-active jobs. It is also useful for starting a service job on still-active jobs for problem determination or to see query optimizer messages. An example CL command to start a service job using the above information would be:

```
STRSRVJOB 039554/QUSER/QRWTSRVR
```

To put the serviced job into debug mode, execute the STRDBG command:

```
STRDBG UPDPROD(*YES)
```

In certain situations the DRDA server saves the joblog of the prestart job before recycling the job and clearing the joblog. This happens when a serious failure has been detected, or when the job ended while being serviced (using the STRSRVJOB command).

To find the saved joblog after the job has ended, issue the following command:

```
WRKJOB userid/QPRTJOB
```

where userid is the name of the userid under which the connection was made (SRR in the above example).

This will display a list of jobs from which one can be selected, or an option menu for a single job. Choose option 4, 'Work with spooled files' to find the saved joblog. It will be the one with file name of QPJOBLOG in case there are multiple files spooled. Option 5 will let you view the joblog file.

An example of the type of query optimizer messages one may see in a server joblog when the job was run under debug is the following:

```
CPI4329    Information   00    03/30/01   16:14:57   QQQIMPLE
            QSYS        3911     QSQOPEN      QSYS         09C4
Message . . . . :   Arrival sequence access was used for file TBL2.
Cause . . . . . :   Arrival sequence access was used to select
  records from member TBL2 of file TBL2 in library SR. If file TBL2
  in library SR is a logical file then member TBL2 of physical file
  TBL2 in library SR is the actual file from which records are
  being selected. A file name of *N for the file indicates it is a
  temporary file. Recovery  . . . :   The use of an access path may
  improve the performance of the query if record selection is
  specified. If an access path does not exist, you may want to
  create one whose left-most key fields match fields in the record
  selection.  Matching more key fields in the access path with
  fields in the record selection will result in improved
  performance. Generally, to force the use of an existing access
  path, specify order by fields that match the left-most key fields
  of that access path. For more information refer to the DB2 for
  iSeries SQL Programming book.
```

Figure 9. A sample query optimizer message

Setup considerations for the DB2 UDB for iSeries DRDA TCP/IP client:

The main consideration for using DB2 UDB for iSeries as a DRDA application requester (AR) over TCP/IP is, besides the security considerations discussed in the next section, adding an RDB directory entry for the remote application server. This is done in a similar manner to what was described in the previous chapter on the use of SNA communications. However, instead of APPC parameters such as remote LU name and transaction program name, there are two TCP/IP parameters: remote host name or IP address, and port number or service name. The second element of the remote location parameter can be specified as *SNA (the default), or *IP (to indicate that the connection will be using TCP/IP).

Security considerations for use of DRDA over TCP/IP:

DRDA over native TCP/IP does not use OS/400® communications security services and concepts such as communications devices, modes, secure location attributes, and conversation security levels which are associated with APPC communications. Therefore, security setup for TCP/IP is quite different.

Two types of security mechanisms are supported by the current DB2 UDB for iSeries implementation of DRDA over TCP/IP:

1. User ID only
2. User ID with password

For a DB2 UDB for iSeries application server (AS), the default security is user ID with password. As the system is installed, inbound TCP/IP connect requests must have a password accompanying the user ID under which the server job is to run. The CHGDDMTCPA command can be used to specify that the password is not

required. To make this change, enter CHGDDMTCPA PWDRQD(*NO). You must have *IOSYSCFG special authority to use this command.

For a DB2 UDB for iSeries application requester (AR), there are two methods that can be used to send a password along with the user ID on TCP/IP connect requests. In the absence of both of these, only a user ID will be sent.

The first way to send a password is to use the USER/USING form of the SQL CONNECT statement. The syntax is:

```
CONNECT TO rdbname USER userid USING 'password'
```

where the lowercase words represent the appropriate connect parameters. In a program using embedded SQL, the userid and password values can be contained in host variables.

The other way that a password can be provided to send on a connect request over TCP/IP is by use of a server authorization entry. Associated with every user profile on the system is a server authorization list. By default the list is empty, but with the ADDSVRAUTE command, entries can be added. When a DRDA connection over TCP/IP is attempted, DB2 UDB for iSeries checks the server authorization list for the user profile under which the client job is running. If a match is found between the RDB name on the CONNECT statement and the SERVER name in an authorization entry, the associated USRID parameter in the entry is used for the connection user ID. If a PASSWORD parameter is stored in the entry, that password is also sent on the connect request.

For a password to be stored using the ADDSVRAUTE command, the QRETSVRSEC system value must to set to '1'. By default, the value is '0'. To make the change, enter:

```
CHGSYSVAL QRETSVRSEC VALUE('1')
```

The syntax of the ADDSVRAUTE command is:

```
ADDSVRAUTE USRPRF(user-profile) SERVER(rdbname) USRID(userid) PASSWORD(password)
```

The USRPRF parameter specifies the user profile under which the application requester job runs. The SERVER parameter specifies the remote RDB name, and the USRID parameter specifies the user profile under which the server job will run. The PASSWORD parameter specifies the password for the user profile at the server.

Note: It is very important that the RDB name in the SERVER parameter be specified in upper case.

If the USRPRF parameter is omitted, it will default to the user profile under which the ADDSVRAUTE command is being run. If the USRID parameter is omitted, it will default to the value of the USRPRF parameter. If the PASSWORD parameter is omitted, or if the QRETSVRSEC value is 0, no password will be stored in the entry. And when a connect attempt is made using the entry, the security mechanism used will be user ID only.

The RMVSVRAUTE command can remove a server authorization entry, and the CHGSVRAUTE command can changed the entry. See the *AS/400 Command Reference* for complete descriptions of these commands.

If a server authorization entry exists for an RDB, and the USER/USING form of the CONNECT statement is also used, the latter takes precedence.

Related concepts:

- "Data representation (iSeries)" on page 119
- "Security considerations for application servers (iSeries)" on page 96
- "Security considerations for application requesters (iSeries)" on page 111
- "DB2 UDB for iSeries" on page 77

Related tasks:

- "Setting up DB2 as an application server using SNA (iSeries)" on page 49
- "Setting up DB2 as an application requester – SNA (iSeries)" on page 31

Chapter 9. Setting up VSE application servers

Setting up DB2 as an application server (VSE)

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 Universal Database for z/OS and OS/390 requester
- A DB2 requester
- A DB2 UDB for iSeries 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.

Procedure:

To establish the network connection to the VSE application server:

1. Establish CICS LU 6.2 sessions to the remote systems
2. Define a VSE application server
3. Prepare and start the DB2 for VSE application server

Related concepts:

- "Security considerations for application servers (VSE)" on page 102
- "DB2 for VSE" on page 88

Related tasks:

- "Establishing CICS LU 6.2 sessions (VSE)" on page 57
- "Defining an application server (VSE)" on page 61
- "Preparing and starting the DB2 application server (VSE)" on page 61

Related reference:

- "Checklist for enabling a DB2 application server (VSE)" on page 127

Setup tasks

Establishing CICS LU 6.2 sessions (VSE)

Establishing CICS LU 6.2 sessions is part of the larger task of setting up DB2 for VSE as an application server. 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.

Procedure:

To establish a CICS LU 6.2 session:

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.
```

Figure 10. Tables to update to enable the CICS Restart Resynchronization capability

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 11 on page 59.

```
         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 11. 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 sync point 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.

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.

   Note: 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.

The *CICS on Open Systems: Intercommunication Guide* contains details on defining and establishing CICS LU 6.2 links with remote systems.

Related tasks:

• "Defining an application server (VSE)" on page 61

Defining an application server (VSE)

Defining a VSE application server is part of the larger task of setting up DB2 for VSE as an application server.

Procedure:

To define a VSE 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 information, see the *DB2 Server for VSE System Administration* and the *DB2 Server for VSE & VM Database Administration*.

Related tasks:

- "Establishing CICS LU 6.2 sessions (VSE)" on page 57
- "Preparing and starting the DB2 application server (VSE)" on page 61

Preparing and starting the DB2 application server (VSE)

Preparing and starting the DB2 for VSE application server is part of the larger task of setting up DB2 for VSE as an application server.

Procedure:

To prepare and start 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 Server for VSE & VM 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.

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.

Related tasks:

- "Establishing CICS LU 6.2 sessions (VSE)" on page 57
- "Defining an application server (VSE)" on page 61

Chapter 10. Setting up VM application servers

Setting up DB2 as an application server (VM)

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 Universal Database for z/OS and OS/390 requester
- A DB2 Universal Database for iSeries requester
- A DB2 for AIX 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.

Procedure:

To process distributed database requests from the DB2 for VM Application Server:

1. Define the application server
2. Prepare the DB2 for VM application requester or application server

Related concepts:

- "Security considerations for application servers (VM)" on page 99
- "DB2 for VM" on page 77
- "Data representation (VM)" on page 122

Related tasks:

- "Defining the application server (VM)" on page 63
- "Preparing the application requester or application server for DRDA communications (VM)" on page 41
- "Setting up DB2 as an application requester (VM)" on page 37

Setup tasks

Defining the application server (VM)

Defining the application server is part of the larger task of setting up DB2 for VM as an application server. For the application server to receive distributed database requests, you define the application server to the local communications subsystem and assign a unique RDB_NAME. The RDB_NAME is provided on the SQLSTART EXEC as the DBNAME parameter.

Procedure:

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. The RDB_NAME you choose for DB2 for VM must be supplied to all users (application requester) 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. 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 12 on page 65 shows a sample RESID names file.

See "Example of a communications directory entry without a password" in the *Security considerations for application requesters (VM)* topic 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.

```
 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 12. Example of a RESID names file

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 on steps on how to post-install load the IBM-supplied servers and file pools see the *VM/ESA Installation Guide*.

For more information on "Using a DRDA Environment", see the *DB2 Server for VM System Administration* book.

Related concepts:

- "Security considerations for application servers (VM)" on page 99
- "Data representation (VM)" on page 122

Part 4. Host and iSeries concepts

Chapter 11. Concepts

DB2 for OS/390 and z/OS

DB2® Universal Database (UDB) for OS/390® and z/OS™ is the IBM® relational database management system for DB2 for OS/390 and z/OS systems. Figure 13 on page 70 shows an OS/390 or z/OS system running a single copy of DB2 UDB for OS/390 and z/OS. It is also possible to run multiple copies of DB2 UDB for OS/390 and z/OS on a single system. To identify copies of DB2 for OS/390 and z/OS within a given system (or copies of DB2 for OS/390 and z/OS within a JES complex), each DB2 system is given a subsystem name, a one- to four- character string unique within the JES complex.

Application requesters:

The application requester connected to a DB2 for OS/390 or z/OS application server can be:

- A DB2 for OS/390 or z/OS requester
- DB2 Connect
- DB2 Universal Database™ Enterprise Server Edition with DB2 Connect™ support enabled.
- 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. Distributed Database Connection Services® (DDCS) Multi-user gateway Version 2.3, DDCS Single-user Version 2.3, and DDCS for Windows Version 2.4 provide this function.
- A DB2 UDB for iSeries™ requester
- An DB2 for VM requester
- Any product that supports the DRDA application requester protocols

Application servers:

The DB2 for OS/390 and z/OS application servers support database access as follows:

- The application requester is permitted to access tables stored at the DB2 for OS/390 and z/OS application server. The application requester must create a package at the DB2 for OS/390 and z/OS application server before the application can be run. The DB2 for OS/390 and z/OS 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 and z/OS 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 Universal Database for z/OS and 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 and z/OS 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.

OS/390 and z/OS address spaces:

In Figure 13, the DB2 for OS/390 and z/OS subsystem name is *xxxx*. Three of the OS/390 and z/OS address space names are prefixed by the DB2 for OS/390 and z/OS subsystem name. These three address spaces make up the DB2 for OS/390 and z/OS product.

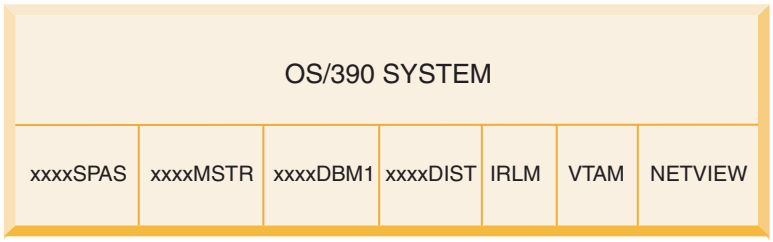

Figure 13. OS/390 and z/OS Address Spaces used by DB2 for OS/390 and z/OS

Figure 13 shows the OS/390 and z/OS address spaces involved in distributed database processing with DB2 for OS/390 and z/OS. These address spaces work together to allow DB2 for OS/390 and z/OS users to access local relational databases and communicate with remote host or iSeries 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 and z/OS product responsible for starting and stopping DB2 for OS/390 and z/OS, and controlling local access to DB2 for OS/390 and z/OS.

xxxxDBM1
The database services address space responsible for accessing relational databases controlled by DB2 for OS/390 and z/OS. 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 and z/OS 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 and z/OS to control access to database resources.

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

NETVIEW
The network management focal point product on OS/390 and z/OS 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.

OS/390 and z/OS attach facilities:

Figure 13 on page 70 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 and z/OS product in one of the following ways:

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

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 and z/OS.

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

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

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

Distributed database connections:

DRDA® defines types of distributed database management system functions. DB2 for OS/390 and z/OS 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 and z/OS 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.

As shown in Figure 14 on page 73, DB2 for OS/390 and z/OS supports three configurations of distributed database connections using two access methods:

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

[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.

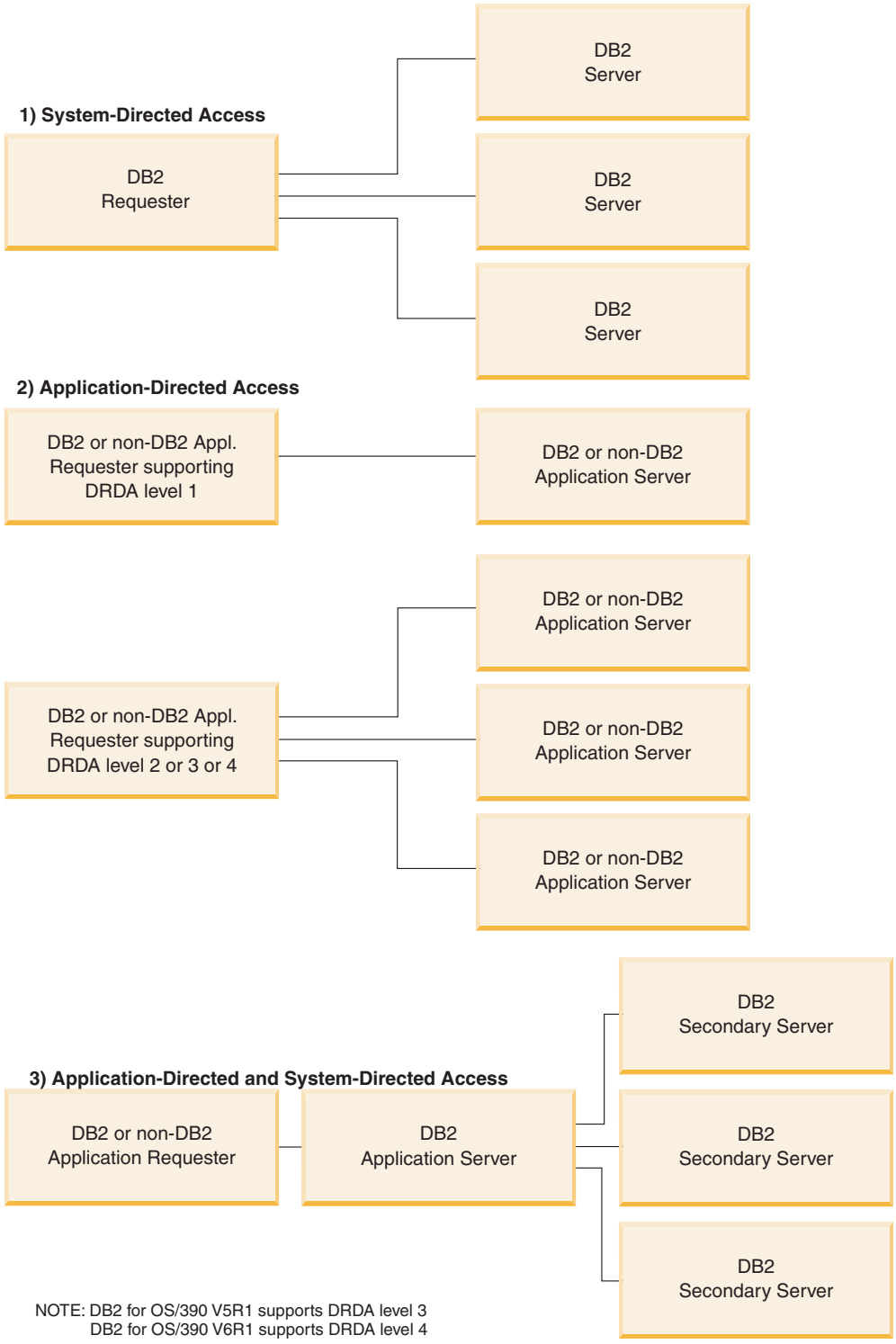

Figure 14. DB2 for OS/390 and z/OS distributed connections

Table 2 on page 74 compares the DB2 for OS/390 and z/OS distributed database connection types.

Table 2. Comparison of DB2 for OS/390 and z/OS 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 and z/OS systems	Can interconnect any two DRDA systems	Application requester can be any DRDA system; servers must be DB2 for OS/390 and z/OS 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 and z/OS 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 and z/OS 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 UDB 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 UDB for OS/390 Version 5.1 introduced 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.

To exploit this, the DB2 Universal Database for z/OS and OS/390 `ZPARM` installation parameter for extended security should be set to the value `YES`. Use the DB2 Universal Database for z/OS and 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`. In the case of an expired password, Windows, UNIX, and Web applications using DB2 Connect will receive error message SQL01404.

TCP/IP security already verified

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

Desktop ODBC and Java™ application security

Workstation ODBC and Java applications use dynamic SQL. This may create security concerns in some installations. DB2 Universal Database for z/OS and 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 and DB2 Connect provide 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`.

Password change support

If an SQL CONNECT statement returns a message indicating that the user ID's password has expired, with DB2 Connect it is possible to change the password without signing on to TSO. Through DRDA, DB2 Universal Database for z/OS and OS/390 can change the password for you.

The old password along with the new password and the verify password must be supplied by the user. If the security specified at the DB2 Connect Enterprise Edition server is `DCS` then a request to change the password is sent to the DB2 Universal Database for z/OS and OS/390 database server. If the security specified is `SERVER` then the password on the DB2 Connect server is changed.

An additional benefit is that a separate LU definition is not required.

Related concepts:

- "Data representation (OS/390 and z/OS)" on page 119
- "Security considerations for application requesters (OS/390 and z/OS)" on page 105
- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Related tasks:

- "Setting up DB2 as an application server (OS/390 and z/OS)" on page 45
- "Setting up DB2 as an application requester (OS/390 and z/OS)" on page 23
- "Setting RU sizes and pacing (OS/390 and z/OS)" on page 76

Subconcepts

Defining communications - SNA (OS/390 and z/OS)

VTAM is the Communications Manager for OS/390 and z/OS systems. VTAM accepts LU 6.2 verbs from DB2 for OS/390 and z/OS and converts these verbs into LU 6.2 data streams you can transmit over the network.

Procedure:

For VTAM to communicate with the partner applications defined in the DB2 for OS/390 and z/OS CDB, you need to provide VTAM with the following information:

- The LU name for each server.

When DB2 for OS/390 and z/OS 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 and z/OS. 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 non-unique 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.

Related concepts:

- "DB2 for OS/390 and z/OS" on page 69

Setting RU sizes and pacing (OS/390 and z/OS)

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.

Procedure:

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.
- 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.

Related concepts:

- "DB2 for OS/390 and z/OS" on page 69

DB2 UDB for iSeries

OS/400 contains DB2® UDB for iSeries, the IBM® relational database management system for iSeries™ systems. DB2 Universal Database for AS/400 Version 4.2 introduced support for DRDA® communications using TCP/IP.

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 UDB for iSeries Query Manager and SQL Development Kit licensed programs to use the DRDA support or to run programs with embedded SQL statements.

Related concepts:

- "Data representation (iSeries)" on page 119
- "Security considerations for application servers (iSeries)" on page 96
- "Security considerations for application requesters (iSeries)" on page 111

Related tasks:

- "Setting up DB2 as an application server using SNA (iSeries)" on page 49
- "Setting up DB2 as an application requester – SNA (iSeries)" on page 31

DB2 for VM

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

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.

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 of 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, refer to the following manuals:

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

Distributed database processing - DRDA and VM components:

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, refer to 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 *VM/ESA CP Programming Services* manual.

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* manual.

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® *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.

As shown in Figure 15, 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.

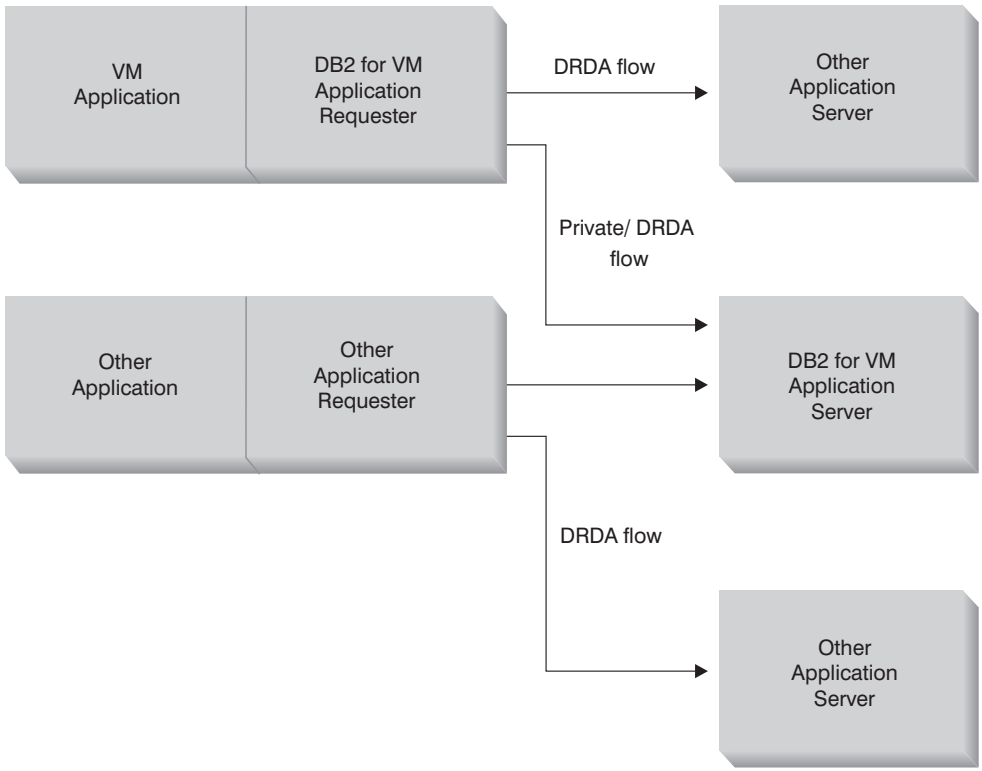

Figure 15. DB2 for VM Application requester and application server

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 [2] 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 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

2. 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.

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 compares functional characteristics of the DB2 for VM application requester SQLINIT processing options.

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 [3]
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

3. 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 (continued)

[SQLDS]	[AUTO]	[DRDA]
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 Server 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 application 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 sync point 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 sync point 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 sync point 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.

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 16 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.

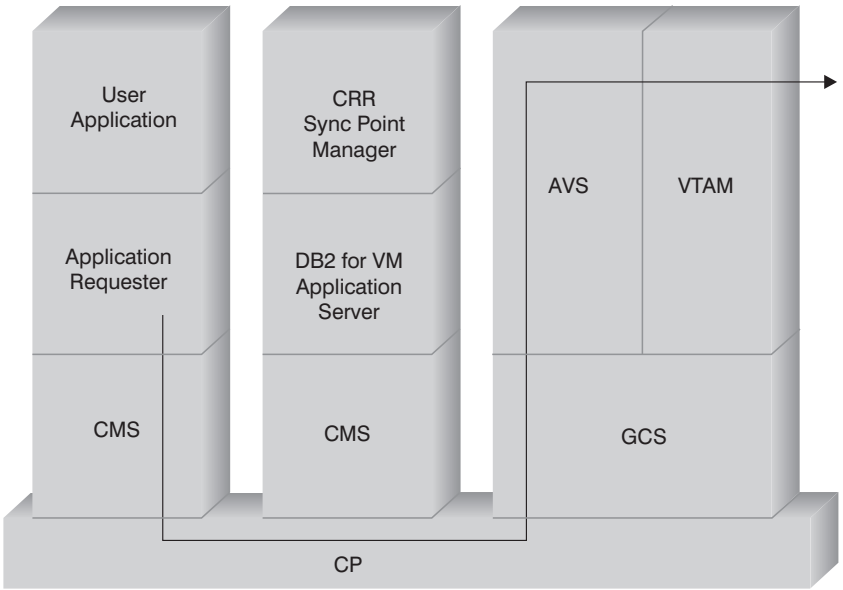

Figure 16. Requesting Access to a Remote Resource

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 17 on page 85 shows that VTAM routes the inbound connection to the specific AVS gateway and then to the application server.

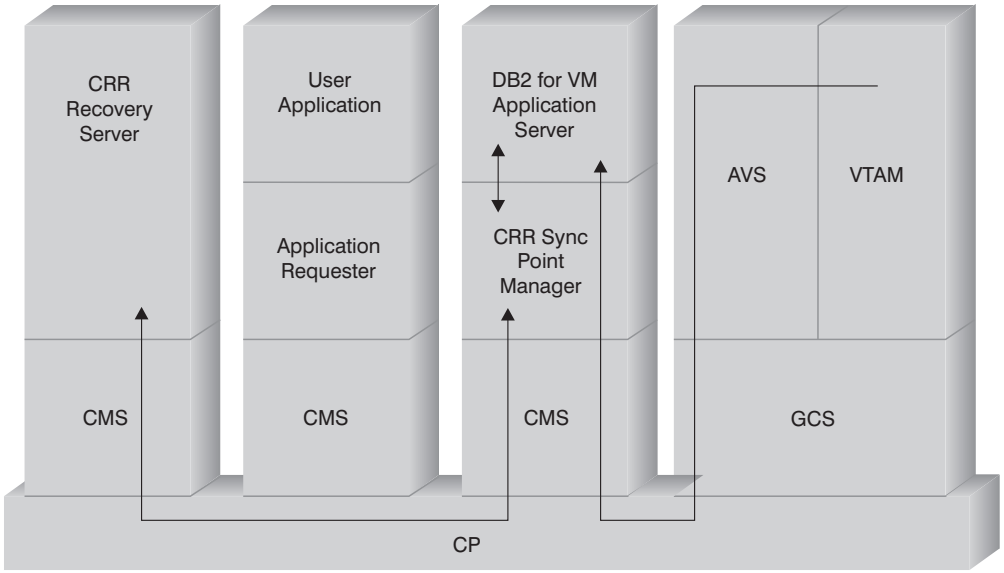

Figure 17. Gaining Access to a Remote Resource

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 (such as, 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

Related concepts:

- “Security considerations for application servers (VM)” on page 99
- “Data representation (VM)” on page 122
- “Security considerations for application requesters (VM)” on page 114
- “DB2 for VSE” on page 88

Related tasks:

- “Setting up DB2 as an application server (VM)” on page 63
- “Setting up DB2 as an application requester (VM)” on page 37

Related reference:

- “Checklist for enabling a DB2 application requester (VM)” on page 128

Subconcepts

Defining communications – application requester (VM)

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 pending 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.

Related concepts:

• "DB2 for VM" on page 77

Setting RU sizes and pacing (VM)

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.

Related concepts:

- "DB2 for VM" on page 77

DB2 for VSE

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

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® sync point management component is integral to DB2 for VSE DRDA distributed unit of work support. It acts as a sync point 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 sync point 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 18 shows how each component plays a role in establishing communications between the DB2 for VSE application server and the remote application requester.

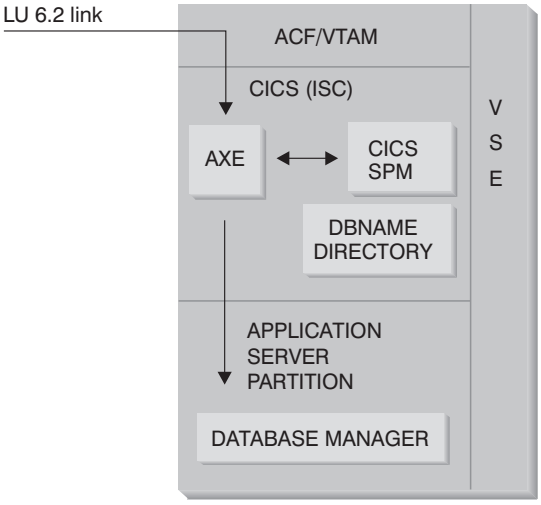

Figure 18. Gaining Access to the application server

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 sync point manager.
- It registers the logical unit of work with the CICS sync point manager.

Application server 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 sync point 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 sync point 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 SPM 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.

Related tasks:

- "Setting up DB2 as an application server (VSE)" on page 57

Chapter 12. Security considerations for application servers

Security considerations for application servers (OS/390 and z/OS)

When an application requester routes a distributed database request to the DB2® for OS/390® and z/OS™ application server, the following security considerations can be involved:

- Come-from checking
- End user names
- Network security
- Database manager security
- Security subsystemSecurity subsystem

Related concepts:

- "Security considerations for application requesters (OS/390 and z/OS)" on page 105
- "DB2 for OS/390 and z/OS" on page 69

Related tasks:

- "Setting up DB2 as an application server (OS/390 and z/OS)" on page 45

Subconcepts

Come-From checking (OS/390 and z/OS)

When the host 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.

Your host system 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.

Related concepts:

- "Security considerations for application servers (OS/390 and z/OS)" on page 91

End user names - application server (OS/390 and z/OS)

The user ID passed by the application requester might not be unique throughout the entire SNA network. The DB2® application server might need to perform inbound name translation to create unique end user names throughout the SNA

network. Similarly, the DB2 application server might need to perform outbound name translation to provide a unique end user name to the secondary servers involved in the application.

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 translates the user ID sent by the application requester and the DB2 plan owner's name (if the application requester is another DB2 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 has authorization exits you can use to provide a list of secondary AUTHIDs and perform additional security checks. See the *DB2 for OS/390 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 resource authorization checks (for example, SQL table privileges) made by DB2 are performed on the translated end user names, rather than on the original user names.

When the DB2 application server receives an end user name from the application requester, several objectives can be accomplished by using the DB2 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', ' ');
```

Figure 19. Changing an end user's name to make it unique

- 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', ' ');
```

Figure 20. Changing an end user's name so that a group of end users are represented by a single name

- 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. 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', ' ', ' ');
```

Figure 21. Restricting the end user names transmitted by an application requester

- You can restrict the application requesters allowed to connect to the DB2 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', ' ', ' ');
```

Figure 22. Restricting the application requesters allowed to connect

Related concepts:

• "Security considerations for application servers (OS/390 and z/OS)" on page 91

Network security - application server (OS/390 and z/OS)

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

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

The only remaining network security consideration is SNA conversation-level security. Some aspects of conversation-level security are unique for a DB2® application server. See the *DB2 for OS/390 Administration Guide* for more details. The DB2 application server plays two distinct roles in network security:

- As a requester to secondary servers, the DB2 application server is responsible for issuing APPC requests that contain the SNA conversation-level security parameters required by the secondary servers. The DB2 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.
- As the server for the application requester, the DB2 application server dictates the SNA conversation level security requirements for the application requester. DB2 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 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 assumes the password is already in RACF® encrypted format (this is only possible for a DB2 application requester). If the ENCRYPTPSWDS column does not contain 'Y', DB2 expects the password in the standard LU 6.2 format (EBCDIC character representation). In either case, DB2 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 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 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 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.

Related concepts:

- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Database manager security - application server (OS/390 and z/OS)

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

- **Packages**— Individual end users are authorized to create, replace, and run packages with the DB2 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 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, 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. 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 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 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 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 connection types can run the package. You can use RACF® and DB2 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 package named MYPKG, which is owned by JOE. JOE can allow SAL to execute the package by issuing the DB2 GRANT USE statement. When SAL executes the package, the following occurs:

- DB2 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.

Related concepts:

- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Security subsystem - application server (OS/390 and z/OS)

The DB2® 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 assumes that the DB2 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 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 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 subsystem.
- In either case, an authorization exit is driven to provide a list of secondary authorization IDs.

For more information, see the *DB2 UDB for OS/390® and z/OS™ Administration Guide*.

Related concepts:

- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Security considerations for application servers (iSeries)

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

- End user names
- Network security parameters

- Database manager security
- iSeries security

End user names:

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

The iSeries 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 iSeries system)

The DB2® UDB for iSeries application server uses session-level security in exactly the same manner as the DB2 UDB for iSeries 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 iSeries 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 the End User Names topic. 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 iSeries 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 iSeries 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 iSeries 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:

Using the **CRTDDMTCPA** command, you can specify 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 iSeries 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 the End User Names topic.

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 *OS/400 Security - Reference*.

Related tasks:

• "Granting and revoking authority (iSeries)" on page 113

Security considerations for application servers (VM)

When an application requester routes a distributed database request to the DB2® for VM application server, the following security considerations can apply:

- End user name
- 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

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
```

Related concepts:

- "DB2 for VM" on page 77
- "Security considerations for application requesters (VM)" on page 114

Related tasks:

- "Setting up DB2 as an application server (VM)" on page 63

Security considerations for application servers (VSE)

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.

- 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 communications directory by setting the appropriate value in the :security tag, as in Figure 23:

```
: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 23. 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.

See the *DB2 Server for VSE System Administration* book for a description of privileged access to the application server by remote application requesters.

See the *CICS on Open Systems: Intercommunication Guide* for how to enable link security.

Related concepts:

- "DB2 for VSE" on page 88

Related tasks:

- "Setting up DB2 as an application server (VSE)" on page 57

Chapter 13. Security considerations for application requesters

Security considerations for application requesters (OS/390 and z/OS)

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:

- End user names
- Network security
- Database manager security
- Security subsystem

Related concepts:

- "DB2 for OS/390 and z/OS" on page 69
- "Security considerations for application servers (OS/390 and z/OS)" on page 91

Related tasks:

- "Setting up DB2 as an application requester (OS/390 and z/OS)" on page 23

Subconcepts

End user names - application requester (OS/390 and z/OS)

On OS/390® and z/OS™ systems, end users are assigned a 1 to 8-character *user ID*. This user ID value must be unique within a particular OS/390 and z/OS 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 situation, JONES in DALLAS can use the privileges granted to JONES at the NEWYORK system.

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

Outbound name translation in DB2 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 authorization is dependent on both the end user's user ID and the user ID of the DB2 for 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.[4]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 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 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 24.

```
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 24. SQL for Outbound Name Translation (SNA)

The resulting CDB tables are shown in Figure 25 on page 107:

4. If the request is being sent to a DB2 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.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
0 0	JONES DSNPLAN	LUDALLAS LUDALLAS	NYJONES	JONESPWD PLANPWD

Figure 25. Outbound Name Translation

Figure 26 shows a more simple example for connecting to a DB2 for OS/390 and z/OS 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 26. SQL for Outbound Name Translation (simple example for SNA).

Figure 27 on page 108 shows a simple example for connecting to a DB2 for OS/390 and z/OS 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 27. SQL for Outbound Name Translation (simple example for TCP/IP).

Related concepts:

- "Security considerations for application requesters (OS/390 and z/OS)" on page 105

Network security - application requester (OS/390 and z/OS)

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.
- 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® 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 obtains the end user's password from two different sources:

- Unencrypted passwords are obtained from the PASSWORD column of the SYSIBM.SYSUSERNAMES table. DB2 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 28 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 28. 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 system. When communicating with a non-DB2 system, do not set ENCRYPTPSWDS to 'Y'.

  DB2 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 29 on page 110 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 29. Sending Encrypted Passwords to Remote Sites

SECURITY=NONE
This option is not supported by DRDA, so DB2 has no provision for this security option.

Related concepts:

- "Security considerations for application requesters (OS/390 and z/OS)" on page 105

Database manager security - application requester (OS/390 and z/OS)

One way the application requester can participate in distributed database security is through outbound name translation. 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® application requester contributes to the distributed system security are:

Binding remote applications
End users bind remote applications at the application server with the DB2 BIND PACKAGE command. DB2 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 plan. DB2 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 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 subsystem.

Executing remote applications
For the DB2 end user to run a remote application, the end user must have authority to run the DB2 plan associated with that application. The DB2 plan owner automatically has authority to run the plan. Other end users can be given authority to run the plan with the DB2 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.

Related concepts:

- "Security considerations for application requesters (OS/390 and z/OS)" on page 105

Security subsystem - application requester (OS/390 and z/OS)

The external security subsystem on MVS™ systems is provided by RACF® and other products that provide an interface compatible with RACF. The DB2® application requester does not have any direct calls to the external security subsystem, with the exception of the encrypted password support. 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 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. As stated earlier, CICS/ESA, TSO, and IMS/ESA® are examples of products that attach end users to DB2.
- If you use SNA session-level security (via the VERIFY keyword on the DB2 VTAM® APPL statement), the external security subsystem is invoked by VTAM to validate the identity of the remote system.

Related concepts:

- "Security considerations for application requesters (OS/390 and z/OS)" on page 105

Security considerations for application requesters (iSeries)

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:

- End user names
- Network security parameters
- Database manager security
- Security enforced by iSeries™ security

End user names:

On iSeries 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 iSeries 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 iSeries 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 iSeries security administrator must verify the security requirements of each application server so they require no more than the iSeries 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 iSeries 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 iSeries application requester.

SECURITY=NONE
Not supported when iSeries is an application requester.

Database manager security:

The iSeries system does not have an external security subsystem. All security is handled through the OS/400 operating system.

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 iSeries application requester or the user ID given in the USER clause of the iSeries 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.

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 the *OS/400 Security - Reference*.

Related concepts:

- "Security considerations for application servers (iSeries)" on page 96
- "DB2 UDB for iSeries" on page 77

Related tasks:

- "Setting up DB2 as an application requester – SNA (iSeries)" on page 31
- "Granting and revoking authority (iSeries)" on page 113

Granting and revoking authority (iSeries)

Procedure:

To grant *USE authority to user USER1 to program PGMA on an iSeries system:

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

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 iSeries 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)
```

Related concepts:

- "Security considerations for application servers (iSeries)" on page 96
- "Security considerations for application requesters (iSeries)" on page 111
- "DB2 UDB for iSeries" on page 77

Security considerations for application requesters (VM)

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 server, the application requester, and the network connecting them. These requirements fall into one or more of the following categories:

- End user names
- 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-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 30, 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 30. 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 Directory 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 31 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 31. 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.

Related concepts:

- "Security considerations for application servers (VM)" on page 99
- "DB2 for VM" on page 77

Related tasks:

- "Setting up DB2 as an application requester (VM)" on page 37

Chapter 14. Data representation

Data representation (OS/390 and z/OS)

DB2® 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, you must set the installation CCSID to the CCSID of the characters generated and sent to DB2 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.

Ensure that your DB2 subsystem has the ability to convert from each application server's CCSID to your DB2 subsystem's installation CCSID. DB2 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 Universal Database™ for OS/390® and z/OS™ Administration Guide* for more information about DB2 UDB for OS/390 and z/OS character conversion.

Related concepts:

- "DB2 for OS/390 and z/OS" on page 69
- "Conversion of character data" in the *Quick Beginnings for DB2 Connect Enterprise Edition*

Related tasks:

- "Setting up DB2 as an application server (OS/390 and z/OS)" on page 45
- "Setting up DB2 as an application requester (OS/390 and z/OS)" on page 23

Data representation (iSeries)

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.

The shipped default CCSID value for the OS/400® is 65535, also referred to as X'FFFF'. This default value is not compatible with the other IBM® products. The system CCSID can be displayed by the CL command DSPSYSVAL QCCSID. It can be changed by the CHGSYSVAL command. For example, CHGSYSVAL QCCSID VALUE(37). The system CCSID can also be overridden by the CCSID associated with the DRDA server job. This CCSID can be set by use of the CHGUSRPRF CL command. For example, CHGUSRPRF MYUSERID CCSID(37).

Application servers:

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® UDB for iSeries™ 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 UDB for iSeries files accessed via DRDA

An SQL table corresponds to an DB2 UDB for iSeries 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. 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.

Application requesters:

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 appropriate choice would be to use the default coded character set identifier for the iSeries 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.

Related concepts:

- "DB2 UDB for iSeries" on page 77
- "Conversion of character data" in the *Quick Beginnings for DB2 Connect Enterprise Edition*

Related tasks:

- "Setting up DB2 as an application server using SNA (iSeries)" on page 49
- "Setting up DB2 as an application requester – SNA (iSeries)" on page 31

Data representation (VM)

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.

Application servers:

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 characters 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 *DB2 Server for VM 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.

Application requesters:

An 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 (£) 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.

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. 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.

See the *DB2 Server for VM System Administration* manual for more detailed information.

Related concepts:

- "DB2 for VM" on page 77
- "DB2 for VSE" on page 88
- "Conversion of character data" in the *Quick Beginnings for DB2 Connect Enterprise Edition*

Related tasks:

- "Setting up DB2 as an application server (VM)" on page 63
- "Setting up DB2 as an application server (VSE)" on page 57
- "Setting up DB2 as an application requester (VM)" on page 37

Part 5. Host and iSeries reference

Chapter 15. Reference

APPC communications products configured using the CA

The Configuration Assistant (CA) can often configure APPC automatically. The following table lists the products that the CA can configure:

Table 4. Products configured using the CA

Products	Platform	Configured by the CA?
IBM Personal Communications V4.2 and later	Windows 98, Windows NT and Windows 2000	Yes
IBM Communications Server (Server)	Windows NT and Windows 2000	Yes
IBM Communications Server (Client)	Windows 98, Windows NT and Windows 2000	No
RUMBA	Windows 98, Windows NT and Windows 2000	Yes
Microsoft SNA (Server)	Windows NT and Windows 2000	No
Microsoft SNA (Client)	Windows 98, Windows NT and Windows 2000	No

Related tasks:

- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11
- "Updating APPC profiles on the DB2 Connect server" on page 12

Checklist for enabling a DB2 application server (VSE)

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.

Related concepts:

- "DB2 for VSE" on page 88

Related tasks:

- "Setting up DB2 as an application server (VSE)" on page 57

Checklist for enabling a DB2 application requester (VM)

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.

Related concepts:

- "DB2 for VM" on page 77

Related tasks:

- "Setting up DB2 as an application requester (VM)" on page 37

TCP/IP parameter value worksheet

As you proceed through the configuration steps, use the *Your Value* column in the following table to record the required values.

Table 5. TCP/IP Values Required at the DB2 Connect Server

Parameter	Description	Sample Value	Your Value

Host name
• Hostname (*hostname*) or
• IP address (*ip_address*)
	Use the *hostname* or *ip_address* of the remote host.
	
	To resolve this parameter:
	• Contact your network administrator to obtain the *hostname*.
	• Contact your network administrator to obtain the *ip_address* or enter the **ping** *hostname* command.
		`nyx`
		or
		`9.21.15.235`	

Service Name
• Connection Service name (*svcename*) or
• Port number/Protocol (*port_number/tcp*)
	Values required in the `services` file.
	
	The Connection Service name is an arbitrary name that represents the connection port number (*port_number*) on the client.
	
	The port number for the DB2 Connect server must be the same as the port number that the *svcename* parameter maps to in the services file at the host database server. (The *svcename* parameter is located in the database manager configuration file on the host.) This value must not be in use by any other applications, and must be unique within the services file.
	
	On UNIX platforms, this value generally must be 1024 or higher.
	
	Contact your database administrator for the values used to configure the host system.
		`host1`
		or
		`3700/tcp`	

Target database name (*target_dbname*)	The database name as it is known on the host or iSeries system. • If you are connecting to a DB2 UDB for OS/390 and z/OS system, use the location name. • If you are connecting to a DB2 UDB for iSeries system, use the local RDB name. • If you are connecting to a DB2 for VM or DB2 for VSE system, use the dbname.	`newyork`	

Table 5. TCP/IP Values Required at the DB2 Connect Server (continued)

Parameter	Description	Sample Value	Your Value
Local database name (*local_dcsname*)	An arbitrary local nickname for use by the DB2 Connect server that represents the remote host or iSeries database.	`ny`	
Node name (*node_name*)	A local alias, or nickname, that describes the node to which you are trying to connect. You can choose any name you want; however, all node name values within your local node directory must be unique.	`db2node`	

Related tasks:

- "Configuring TCP/IP communications manually between DB2 Connect and a host and iSeries database server" on page 3

TCP/IP parameter values for cataloging databases

Fill in the *Your Value* column in the following worksheet.

Table 6. Worksheet: Parameter Values for Cataloging Databases

Parameter	Description	Sample Value	Your Value
Database name (*database_name*)	The local DCS database name (*local_dcsname*) of the *remote* database, you specified this when you catalogued the DCS database directory, for example, `ny`.	`ny`	
Database alias (*database_alias*)	An arbitrary local nickname for the remote database. If you do not provide one, the default is the same as the database name (*database_name*). Use this name when you connect to the database from a client.	`localny`	
Node name (*node_name*)	Use the same value for the Node name (*node_name*) that you used to catalog the node in.	`db2node`	

Related tasks:

- "Configuring TCP/IP communications manually between DB2 Connect and a host and iSeries database server" on page 3
- "Cataloging the database" on page 7

APPC parameter value worksheet

Before you configure the DB2 Connect server, have your host or iSeries administrator and LAN administrator fill in copies of this worksheet for *each* host or iSeries database to which you want to connect.

After you fill in the entries in the *Your Value* column, you can use the worksheet to configure APPC communications for DB2 Connect. During the configuration process, replace the sample values that appear in the configuration instructions

with your values from the worksheet. Use the boxed numbers (for example, 1) to relate the configuration instructions to the worksheet values.

The worksheet and configuration instructions supply suggested or sample values for required configuration parameters. For other parameters, use the communications program's default values. If your network configuration is different from these instructions, consult your Network Administrator for values that are appropriate to your network.

In the configuration instructions, the * symbol denotes entries that need to be changed but do not have a representation on the worksheet.

Table 7. Worksheet for planning host and iSeries server connections

Ref.	Name at the DB2 Connect server	Network or VTAM Name	Sample Value	Your Value
Network Elements at the host or iSeries database server				
1	Host name	Local network name	SPIFNET	
2	Partner LU name	Application name	NYM2DB2	
3	Network ID		SPIFNET	
4	Partner node name	Local CP or SSCP name	NYX	
5	Target database name (*target_dbname*)	**OS/390 or z/OS:** LOCATION NAME **VM/VSE:** DBNAME **iSeries:** RDB name	NEWYORK	
6	Link name or mode name		IBMRDB	
7	Connection name (link name)		LINKHOST	
8	Remote network or LAN address	Local adapter or destination address	400009451902	
Network Elements at the DB2 Connect server				
9	Network or LAN ID		SPIFNET	
10	Local control point name		NYX1GW	
11	Local LU name		NYX1GW0A	
12	Local LU alias		NYX1GW0A	
13	Local node or node ID	ID BLK	071	
14		ID NUM	27509	
15	Mode name		IBMRDB	
16	Symbolic destination name		DB2CPIC	

Table 7. Worksheet for planning host and iSeries server connections (continued)

Ref.	Name at the DB2 Connect server	Network or VTAM Name	Sample Value	Your Value
17	Remote Transaction program (TP) name		**OS/390 or z/OS:** X'07'6DB ('07F6C4C2') or DB2DRDA **VM/VSE:** AXE for VSE. The DB2 for VM db name, or X'07'6DB ('07F6C4C2') for VM **iSeries:** X'07'6DB ('07F6C4C2') or QCNTEDDM	
DB2 Directory Entries at the DB2 Connect server				
19	Node name		db2node	
19	Security		program	
20	Local database name (*local_dcsname*)		ny	

For each server that you are connecting to, fill in a copy of the worksheet as follows:

1. For *network ID*, determine the network name of both the host or iSeries, and the DB2 Connect servers (1 , 3 , and 9). Usually these values will be the same. For example, SPIFNET.
2. For the *partner LU name* (2), determine the VTAM application (APPL) name for OS/390, z/OS, VSE, or VM. Determine the local CP name for iSeries.
3. For *partner node name* (4), determine the System Services Control Point (SSCP) name for OS/390, z/OS, VM, or VSE. Determine the local control point name for an iSeries.
4. For *database name* (5), determine the name of the host and iSeries database. This is the *LOCATION NAME* for OS/390 or z/OS, the *DBNAME* for VM or VSE, or a relational database (RDB) name for iSeries.
5. For *mode name* (6 and 15), usually the default IBMDRB is sufficient.
6. For *remote network address* (8), determine the controller address or local adapter address of the target host or iSeries system.
7. Determine the *local control point name* (10) of the DB2 Connect server. This is usually the same as the PU name for the system.
8. Determine the *local LU name* that DB2 Connect will use (11). If you use a sync point manager (SPM) to manage multisite updates (two-phase commit), the local LU should be the LU used for the SPM. In this case, that LU cannot also be the control point LU.
9. For *local LU alias* (12), you usually use the same value as for the local LU name (11).
10. For *local node* or *node ID* (13 plus 14), determine the IDBLK and IDNUM of the DB2 Connect server. The default value should be correct.
11. For *symbolic destination name* (16), choose a suitable value.
12. For (remote) *transaction program (TP) name* (17), use the defaults listed in the worksheet.

13. Leave the other items blank for now (**18** to **21**).

Related tasks:

- "Configuring APPC communications manually between DB2 Connect and a host and iSeries database server" on page 11

DB2 Connect VTAM APPL statement keywords

Many keywords are available on the VTAM APPL statement. The keywords discussed here address topics in this book.

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.

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 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 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 communications database.

DSESLIM=20
The total number of sessions (winner and loser sessions) you can establish between DB2 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 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 mode name exists.

PRTCT=PSWDBD1
Identifies the VTAM password to use when DB2 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 change log inventory utility.

SECACPT=ALREADYV
Identifies the highest SNA conversation-level security value accepted by this DB2 system when it receives a distributed database request from a remote system. The ALREADYV keyword indicates this DB2 system can accept three SNA session security options from other DRDA systems that request data from this DB2 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 rejects DRDA requests that specify SECURITY=NONE.

It is best to always specify SECACPT=ALREADYV, because the SNA conversation security level for each DB2 partner is taken from the DB2 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 system. The NONE value indicates that partner LU verification is not required.

DB2 does not restrict your choice for the VERIFY keyword. In an untrusted 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 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 automatically uses two-phase commit if the partner can support it.

ATNLOSS=ALL
Indicates that DB2 needs to be informed each time a VTAM session ends. This ensures that DB2 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 Windows 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
```

Related concepts:

• "Security subsystem - application server (OS/390 and z/OS)" on page 96

- “Network security - application server (OS/390 and z/OS)” on page 94
- “Network security - application requester (OS/390 and z/OS)” on page 108
- “Security subsystem - application requester (OS/390 and z/OS)” on page 111

Related tasks:
- “Setting up DB2 as an application server (OS/390 and z/OS)” on page 45
- “Setting up DB2 as an application requester (OS/390 and z/OS)” on page 23

Part 6. Appendixes

Appendix A. DB2 Universal Database technical information

DB2 documentation and help

DB2® technical information is available through the following tools and methods:

- DB2 Information Center
  - Topics
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- Downloadable PDF files, PDF files on CD, and printed books
  - Guides
  - Reference manuals
- Command line help
  - Command help
  - Message help
  - SQL state help
- Installed source code
  - Sample programs

You can access additional DB2 Universal Database™ technical information such as technotes, white papers, and Redbooks™ online at ibm.com®. Access the DB2 Information Management software library site at www.ibm.com/software/data/pubs/.

DB2 documentation updates

IBM® may periodically make documentation FixPaks and other documentation updates to the DB2 Information Center available. If you access the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/, you will always be viewing the most up-to-date information. If you have installed the DB2 Information Center locally, then you need to install any updates manually before you can view them. Documentation updates allow you to update the information that you installed from the *DB2 Information Center CD* when new information becomes available.

The Information Center is updated more frequently than either the PDF or the hardcopy books. To get the most current DB2 technical information, install the documentation updates as they become available or go to the DB2 Information Center at the www.ibm.com site.

DB2 Information Center

The DB2® Information Center gives you access to all of the information you need to take full advantage of DB2 family products, including DB2 Universal Database™, DB2 Connect™, DB2 Information Integrator and DB2 Query Patroller™. The DB2 Information Center also contains information for major DB2 features and components including replication, data warehousing, and the DB2 extenders.

The DB2 Information Center has the following features if you view it in Mozilla 1.0 or later or Microsoft® Internet Explorer 5.5 or later. Some features require you to enable support for JavaScript™:

Flexible installation options
You can choose to view the DB2 documentation using the option that best meets your needs:

- To effortlessly ensure that your documentation is always up to date, you can access all of your documentation directly from the DB2 Information Center hosted on the IBM® Web site at http://publib.boulder.ibm.com/infocenter/db2help/
- To minimize your update efforts and keep your network traffic within your intranet, you can install the DB2 documentation on a single server on your intranet
- To maximize your flexibility and reduce your dependence on network connections, you can install the DB2 documentation on your own computer

Search
You can search all of the topics in the DB2 Information Center by entering a search term in the **Search** text field. You can retrieve exact matches by enclosing terms in quotation marks, and you can refine your search with wildcard operators (*, ?) and Boolean operators (AND, NOT, OR).

Task-oriented table of contents
You can locate topics in the DB2 documentation from a single table of contents. The table of contents is organized primarily by the kind of tasks you may want to perform, but also includes entries for product overviews, goals, reference information, an index, and a glossary.

- Product overviews describe the relationship between the available products in the DB2 family, the features offered by each of those products, and up to date release information for each of these products.
- Goal categories such as installing, administering, and developing include topics that enable you to quickly complete tasks and develop a deeper understanding of the background information for completing those tasks.
- Reference topics provide detailed information about a subject, including statement and command syntax, message help, and configuration parameters.

Show current topic in table of contents
You can show where the current topic fits into the table of contents by clicking the **Refresh / Show Current Topic** button in the table of contents frame or by clicking the **Show in Table of Contents** button in the content frame. This feature is helpful if you have followed several links to related topics in several files or arrived at a topic from search results.

Index You can access all of the documentation from the index. The index is organized in alphabetical order by index term.

Glossary
You can use the glossary to look up definitions of terms used in the DB2 documentation. The glossary is organized in alphabetical order by glossary term.

Integrated localized information
The DB2 Information Center displays information in the preferred

language set in your browser preferences. If a topic is not available in your preferred language, the DB2 Information Center displays the English version of that topic.

For iSeries™ technical information, refer to the IBM eServer™ iSeries information center at www.ibm.com/eserver/iseries/infocenter/.

Related tasks:

- "Updating the DB2 Information Center installed on your computer or intranet server" on page 148

DB2 Information Center installation scenarios

Different working environments can pose different requirements for how to access DB2® information. The DB2 Information Center can be accessed on the IBM® Web site, on a server on your organization's network, or on a version installed on your computer. In all three cases, the documentation is contained in the DB2 Information Center, which is an architected web of topic-based information that you view with a browser. By default, DB2 products access the DB2 Information Center on the IBM Web site. However, if you want to access the DB2 Information Center on an intranet server or on your own computer, you must install the DB2 Information Center using the DB2 Information Center CD found in your product Media Pack. Refer to the summary of options for accessing DB2 documentation which follows, along with the three installation scenarios, to help determine which method of accessing the DB2 Information Center works best for you and your work environment, and what installation issues you might need to consider.

Summary of options for accessing DB2 documentation:

The following table provides recommendations on which options are possible in your work environment for accessing the DB2 product documentation in the DB2 Information Center.

Internet access	Intranet access	Recommendation
Yes	Yes	Access the DB2 Information Center on the IBM Web site, or access the DB2 Information Center installed on an intranet server.
Yes	No	Access the DB2 Information Center on the IBM Web site.
No	Yes	Access the DB2 Information Center installed on an intranet server.
No	No	Access the DB2 Information Center on a local computer.

Scenario: Accessing the DB2 Information Center on your computer:

Tsu-Chen owns a factory in a small town that does not have a local ISP to provide him with Internet access. He purchased DB2 Universal Database™ to manage his inventory, his product orders, his banking account information, and his business expenses. Never having used a DB2 product before, Tsu-Chen needs to learn how to do so from the DB2 product documentation.

After installing DB2 Universal Database on his computer using the typical installation option, Tsu-Chen tries to access the DB2 documentation. However, his browser gives him an error message that the page he tried to open cannot be

found. Tsu-Chen checks the installation manual for his DB2 product and discovers that he has to install the DB2 Information Center if he wants to access DB2 documentation on his computer. He finds the *DB2 Information Center CD* in the media pack and installs it.

From the application launcher for his operating system, Tsu-Chen now has access to the DB2 Information Center and can learn how to use his DB2 product to increase the success of his business.

Scenario: Accessing the DB2 Information Center on the IBM Web site:

Colin is an information technology consultant with a training firm. He specializes in database technology and SQL and gives seminars on these subjects to businesses all over North America using DB2 Universal Database. Part of Colin's seminars includes using DB2 documentation as a teaching tool. For example, while teaching courses on SQL, Colin uses the DB2 documentation on SQL as a way to teach basic and advanced syntax for database queries.

Most of the businesses at which Colin teaches have Internet access. This situation influenced Colin's decision to configure his mobile computer to access the DB2 Information Center on the IBM Web site when he installed the latest version of DB2 Universal Database. This configuration allows Colin to have online access to the latest DB2 documentation during his seminars.

However, sometimes while travelling Colin does not have Internet access. This posed a problem for him, especially when he needed to access to DB2 documentation to prepare for seminars. To avoid situations like this, Colin installed a copy of the DB2 Information Center on his mobile computer.

Colin enjoys the flexibility of always having a copy of DB2 documentation at his disposal. Using the **db2set** command, he can easily configure the registry variables on his mobile computer to access the DB2 Information Center on either the IBM Web site, or his mobile computer, depending on his situation.

Scenario: Accessing the DB2 Information Center on an intranet server:

Eva works as a senior database administrator for a life insurance company. Her administration responsibilities include installing and configuring the latest version of DB2 Universal Database on the company's UNIX® database servers. Her company recently informed its employees that, for security reasons, it would not provide them with Internet access at work. Because her company has a networked environment, Eva decides to install a copy of the DB2 Information Center on an intranet server so that all employees in the company who use the company's data warehouse on a regular basis (sales representatives, sales managers, and business analysts) have access to DB2 documentation.

Eva instructs her database team to install the latest version of DB2 Universal Database on all of the employee's computers using a response file, to ensure that each computer is configured to access the DB2 Information Center using the host name and the port number of the intranet server.

However, through a misunderstanding Migual, a junior database administrator on Eva's team, installs a copy of the DB2 Information Center on several of the employee computers, rather than configuring DB2 Universal Database to access the DB2 Information Center on the intranet server. To correct this situation Eva tells Migual to use the **db2set** command to change the DB2 Information Center registry

variables (DB2_DOCHOST for the host name, and DB2_DOCPORT for the port number) on each of these computers. Now all of the appropriate computers on the network have access to the DB2 Information Center, and employees can find answers to their DB2 questions in the DB2 documentation.

Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)

DB2 product documentation can be accessed in three ways: on the IBM Web site, on an intranet server, or on a version installed on your computer. By default, DB2 products access DB2 documentation on the IBM Web site. If you want to access the DB2 documentation on an intranet server or on your own computer, you must install the documentation from the *DB2 Information Center CD*. Using the DB2 Setup wizard, you can define your installation preferences and install the DB2 Information Center on a computer that uses a UNIX operating system.

Prerequisites:

This section lists the hardware, operating system, software, and communication requirements for installing the DB2 Information Center on UNIX computers.

- **Hardware requirements**

  You require one of the following processors:
  - PowerPC (AIX)
  - HP 9000 (HP-UX)
  - Intel 32–bit (Linux)
  - Solaris UltraSPARC computers (Solaris Operating Environment)
- **Operating system requirements**

  You require one of the following operating systems:
  - IBM AIX 5.1 (on PowerPC)
  - HP-UX 11i (on HP 9000)
  - Red Hat Linux 8.0 (on Intel 32–bit)
  - SuSE Linux 8.1 (on Intel 32–bit)
  - Sun Solaris Version 8 (on Solaris Operating Environment UltraSPARC computers)

  Note: The DB2 Information Center runs on a subset of the UNIX operating systems on which DB2 clients are supported. It is therefore recommended that you either access the DB2 Information Center from the IBM Web site, or that you install and access the DB2 Information Center on an intranet server.
- **Software requirements**
  - The following browser is supported:
    - Mozilla Version 1.0 or greater
- The DB2 Setup wizard is a graphical installer. You must have an implementation of the X Window System software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your computer. Before you can run the DB2 Setup wizard you must ensure that you have properly exported your display. For example, enter the following command at the command prompt:

  ```
  export DISPLAY=9.26.163.144:0.
  ```
- **Communication requirements**

– TCP/IP

Procedure:

To install the DB2 Information Center using the DB2 Setup wizard:

1. Log on to the system.
2. Insert and mount the DB2 Information Center product CD on your system.
3. Change to the directory where the CD is mounted by entering the following command:

   ```
   cd /cd
   ```

   where */cd* represents the mount point of the CD.
4. Enter the **./db2setup** command to start the DB2 Setup wizard.
5. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation of the DB2 Information Center, click **Install Product**. Online help is available to guide you through the remaining steps. To invoke the online help, click **Help**. You can click **Cancel** at any time to end the installation.
6. On the **Select the product you would like to install** page, click **Next**.
7. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup wizard will guide you through the program setup process.
8. To proceed with the installation, you must accept the license agreement. On the **License Agreement** page, select **I accept the terms in the license agreement** and click **Next**.
9. Select **Install DB2 Information Center on this computer** on the **Select the installation action** page. If you want to use a response file to install the DB2 Information Center on this or other computers at a later time, select **Save your settings in a response file**. Click **Next**.
10. Select the languages in which the DB2 Information Center will be installed on **Select the languages to install** page. Click **Next**.
11. Configure the DB2 Information Center for incoming communication on the **Specify the DB2 Information Center port** page. Click **Next** to continue the installation.
12. Review the installation choices you have made in the **Start copying files** page. To change any settings, click **Back**. Click **Install** to copy the DB2 Information Center files onto your computer.

You can also install the DB2 Information Center using a response file.

The installation logs `db2setup.his`, `db2setup.log`, and `db2setup.err` are located, by default, in the `/tmp` directory.

The `db2setup.log` file captures all DB2 product installation information, including errors. The `db2setup.his` file records all DB2 product installations on your computer. DB2 appends the `db2setup.log` file to the `db2setup.his` file. The `db2setup.err` file captures any error output that is returned by Java, for example, exceptions and trap information.

When the installation is complete, the DB2 Information Center will be installed in one of the following directories, depending upon your UNIX operating system:

- AIX: /usr/opt/db2_08_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1

- Solaris Operating Environment: /opt/IBM/db2/V8.1

Related tasks:

- "Installing the DB2 Information Center using the DB2 Setup wizard (Windows)" on page 145

Installing the DB2 Information Center using the DB2 Setup wizard (Windows)

DB2 product documentation can be accessed in three ways: on the IBM Web site, on an intranet server, or on a version installed on your computer. By default, DB2 products access DB2 documentation on the IBM Web site. If you want to access the DB2 documentation on an intranet server or on your own computer, you must install the DB2 documentation from the *DB2 Information Center CD*. Using the DB2 Setup wizard, you can define your installation preferences and install the DB2 Information Center on a computer that uses a Windows operating system.

Prerequisites:

This section lists the hardware, operating system, software, and communication requirements for installing the DB2 Information Center on Windows.

- **Hardware requirements**

  You require one of the following processors:

  – 32-bit computers: a Pentium or Pentium compatible CPU
- **Operating system requirements**

  You require one of the following operating systems:

  – Windows 2000

  – Windows XP

  Note: The DB2 Information Center runs on a subset of the Windows operating systems on which DB2 clients are supported. It is therefore recommended that you either access the DB2 Information Center on the IBM Web site, or that you install and access the DB2 Information Center on an intranet server.
- **Software requirements**

  – The following browsers are supported:

    - Mozilla 1.0 or greater

    - Internet Explorer Version 5.5 or 6.0 (Version 6.0 for Windows XP)
- **Communication requirements**

  – TCP/IP

Restrictions:

- You require an account with administrative privileges to install the DB2 Information Center.

Procedure:

To install the DB2 Information Center using the DB2 Setup wizard:

1. Log on to the system with the account that you have defined for the DB2 Information Center installation.

2. Insert the CD into the drive. If enabled, the auto-run feature starts the IBM DB2 Setup Launchpad.
3. The DB2 Setup wizard determines the system language and launches the setup program for that language. If you want to run the setup program in a language other than English, or the setup program fails to auto-start, you can start the DB2 Setup wizard manually.

   To start the DB2 Setup wizard manually:

   a. Click **Start** and select **Run**.
   b. In the **Open** field, type the following command:

      ```
      x:\setup.exe /i 2-letter language identifier
      ```

      where *x:* represents your CD drive, and *2-letter language identifier* represents the language in which the setup program will be run.
   c. Click **OK**.
4. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation of the DB2 Information Center, click **Install Product**. Online help is available to guide you through the remaining steps. To invoke the online help, click **Help**. You can click **Cancel** at any time to end the installation.
5. On the **Select the product you would like to install** page, click **Next**.
6. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup wizard will guide you through the program setup process.
7. To proceed with the installation, you must accept the license agreement. On the **License Agreement** page, select **I accept the terms in the license agreement** and click **Next**.
8. Select **Install DB2 Information Center on this computer** on the **Select the installation action** page. If you want to use a response file to install the DB2 Information Center on this or other computers at a later time, select **Save your settings in a response file**. Click **Next**.
9. Select the languages in which the DB2 Information Center will be installed on **Select the languages to install** page. Click **Next**.
10. Configure the DB2 Information Center for incoming communication on the **Specify the DB2 Information Center port** page. Click **Next** to continue the installation.
11. Review the installation choices you have made in the **Start copying files** page. To change any settings, click **Back**. Click **Install** to copy the DB2 Information Center files onto your computer.

You can install the DB2 Information Center using a response file. You can also use the **db2rspgn** command to generate a response file based on an existing installation.

For information on errors encountered during installation, see the `db2.log` and `db2wi.log` files located in the `'My Documents'\DB2LOG\` directory. The location of the `'My Documents'` directory will depend on the settings on your computer.

The `db2wi.log` file captures the most recent DB2 installation information. The `db2.log` captures the history of DB2 product installations.

Related tasks:

- "Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)" on page 143

Invoking the DB2 Information Center

The DB2 Information Center gives you access to all of the information that you need to use DB2 products for Linux, UNIX, and Windows operating systems such as DB2 Universal Database, DB2 Connect, DB2 Information Integrator, and DB2 Query Patroller.

You can invoke the DB2 Information Center from one of the following places:

- Computers on which a DB2 UDB client or server is installed
- An intranet server or local computer on which the DB2 Information Center installed
- The IBM Web site

Prerequisites:

Before you invoke the DB2 Information Center:

- *Optional*: Configure your browser to display topics in your preferred language
- *Optional*: Configure your DB2 client to use the DB2 Information Center installed on your computer or intranet server

Procedure:

To invoke the DB2 Information Center on a computer on which a DB2 UDB client or server is installed:

- From the Start Menu (Windows operating system): Click **Start → Programs → IBM DB2 → Information → Information Center**.
- From the command line prompt:
  - For Linux and UNIX operating systems, issue the **db2icdocs** command.
  - For the Windows operating system, issue the **db2icdocs.exe** command.

To open the DB2 Information Center installed on an intranet server or local computer in a Web browser:

- Open the Web page at http://<host-name>:<port-number>/, where <host-name> represents the host name and <port-number> represents the port number on which the DB2 Information Center is available.

To open the DB2 Information Center on the IBM Web site in a Web browser:

- Open the Web page at publib.boulder.ibm.com/infocenter/db2help/.

Related concepts:

- "DB2 Information Center" on page 139

Related tasks:

- "Displaying topics in your preferred language in the DB2 Information Center" on page 148
- "Invoking contextual help from a DB2 tool" on page 155
- "Updating the DB2 Information Center installed on your computer or intranet server" on page 148
- "Invoking message help from the command line processor" on page 156
- "Invoking command help from the command line processor" on page 156
- "Invoking SQL state help from the command line processor" on page 157

Updating the DB2 Information Center installed on your computer or intranet server

The DB2 Information Center available from http://publib.boulder.ibm.com/infocenter/db2help/ will be periodically updated with new or changed documentation. IBM may also make DB2 Information Center updates available to download and install on your computer or intranet server. Updating the DB2 Information Center does not update DB2 client or server products.

Prerequisites:

You must have access to a computer that is connected to the Internet.

Procedure:

To update the DB2 Information Center installed on your computer or intranet server:

1. Open the DB2 Information Center hosted on the IBM Web site at: http://publib.boulder.ibm.com/infocenter/db2help/
2. In the Downloads section of the welcome page under the Service and Support heading, click the **DB2 Universal Database documentation** link.
3. Determine if the version of your DB2 Information Center is out of date by comparing the latest refreshed documentation image level to the documentation level you have installed. The documentation level you have installed is listed on the DB2 Information Center welcome page.
4. If a more recent version of the DB2 Information Center is available, download the latest refreshed *DB2 Information Center* image applicable to your operating system.
5. To install the refreshed *DB2 Information Center* image, follow the instructions provided on the Web page.

Related tasks:

- "Copying files from the DB2 HTML Documentation CD to a Web server" in the *Quick Beginnings for DB2 Personal Edition*

Related reference:

- "DB2 PDF and printed documentation" on page 149

Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

Procedure:

To display topics in your preferred language in the Internet Explorer browser:

1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.

2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button.

     Note: Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Refresh the page to display the DB2 Information Center in your preferred language.

To display topics in your preferred language in the Mozilla browser:

1. In Mozilla, select the **Edit** —> **Preferences** —> **Languages** button. The Languages panel is displayed in the Preferences window.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Refresh the page to display the DB2 Information Center in your preferred language.

DB2 PDF and printed documentation

The following tables provide official book names, form numbers, and PDF file names. To order hardcopy books, you must know the official book name. To print a PDF file, you must know the PDF file name.

The DB2 documentation is categorized by the following headings:

- Core DB2 information
- Administration information
- Application development information
- Business intelligence information
- DB2 Connect information
- Getting started information
- Tutorial information
- Optional component information
- Release notes

The following tables describe, for each book in the DB2 library, the information needed to order the hard copy, or to print or view the PDF for that book. A full description of each of the books in the DB2 library is available from the IBM Publications Center at www.ibm.com/shop/publications/order

Core DB2 information

The information in these books is fundamental to all DB2 users; you will find this information useful whether you are a programmer, a database administrator, or someone who works with DB2 Connect, DB2 Warehouse Manager, or other DB2 products.

Table 8. Core DB2 information

Name	Form Number	PDF File Name
IBM DB2 Universal Database Command Reference	SC09-4828	db2n0x81
IBM DB2 Universal Database Glossary	No form number	db2t0x81
IBM DB2 Universal Database Message Reference, Volume 1	GC09-4840, not available in hardcopy	db2m1x81
IBM DB2 Universal Database Message Reference, Volume 2	GC09-4841, not available in hardcopy	db2m2x81
IBM DB2 Universal Database What's New	SC09-4848	db2q0x81

Administration information

The information in these books covers those topics required to effectively design, implement, and maintain DB2 databases, data warehouses, and federated systems.

Table 9. Administration information

Name	Form number	PDF file name
IBM DB2 Universal Database Administration Guide: Planning	SC09-4822	db2d1x81
IBM DB2 Universal Database Administration Guide: Implementation	SC09-4820	db2d2x81
IBM DB2 Universal Database Administration Guide: Performance	SC09-4821	db2d3x81
IBM DB2 Universal Database Administrative API Reference	SC09-4824	db2b0x81
IBM DB2 Universal Database Data Movement Utilities Guide and Reference	SC09-4830	db2dmx81
IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference	SC09-4831	db2hax81
IBM DB2 Universal Database Data Warehouse Center Administration Guide	SC27-1123	db2ddx81
IBM DB2 Universal Database SQL Reference, Volume 1	SC09-4844	db2s1x81
IBM DB2 Universal Database SQL Reference, Volume 2	SC09-4845	db2s2x81
IBM DB2 Universal Database System Monitor Guide and Reference	SC09-4847	db2f0x81

Application development information

The information in these books is of special interest to application developers or programmers working with DB2 Universal Database (DB2 UDB). You will find information about supported languages and compilers, as well as the

documentation required to access DB2 UDB using the various supported programming interfaces, such as embedded SQL, ODBC, JDBC, SQLJ, and CLI. If you are using the DB2 Information Center, you can also access HTML versions of the source code for the sample programs.

Table 10. Application development information

Name	Form number	PDF file name
IBM DB2 Universal Database Application Development Guide: Building and Running Applications	SC09-4825	db2axx81
IBM DB2 Universal Database Application Development Guide: Programming Client Applications	SC09-4826	db2a1x81
IBM DB2 Universal Database Application Development Guide: Programming Server Applications	SC09-4827	db2a2x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1	SC09-4849	db2l1x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2	SC09-4850	db2l2x81
IBM DB2 Universal Database Data Warehouse Center Application Integration Guide	SC27-1124	db2adx81
IBM DB2 XML Extender Administration and Programming	SC27-1234	db2sxx81

Business intelligence information

The information in these books describes how to use components that enhance the data warehousing and analytical capabilities of DB2 Universal Database.

Table 11. Business intelligence information

Name	Form number	PDF file name
IBM DB2 Warehouse Manager Standard Edition Information Catalog Center Administration Guide	SC27-1125	db2dix81
IBM DB2 Warehouse Manager Standard Edition Installation Guide	GC27-1122	db2idx81
IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager	SC18-7727	iwhe1mstx80

DB2 Connect information

The information in this category describes how to access data on mainframe and midrange servers using DB2 Connect Enterprise Edition or DB2 Connect Personal Edition.

Table 12. DB2 Connect information

Name	Form number	PDF file name
IBM Connectivity Supplement	No form number	db2h1x81
IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition	GC09-4833	db2c6x81
IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition	GC09-4834	db2c1x81
IBM DB2 Connect User's Guide	SC09-4835	db2c0x81

Getting started information

The information in this category is useful when you are installing and configuring servers, clients, and other DB2 products.

Table 13. Getting started information

Name	Form number	PDF file name
IBM DB2 Universal Database Quick Beginnings for DB2 Clients	GC09-4832, not available in hardcopy	db2itx81
IBM DB2 Universal Database Quick Beginnings for DB2 Servers	GC09-4836	db2isx81
IBM DB2 Universal Database Quick Beginnings for DB2 Personal Edition	GC09-4838	db2i1x81
IBM DB2 Universal Database Installation and Configuration Supplement	GC09-4837, not available in hardcopy	db2iyx81
IBM DB2 Universal Database Quick Beginnings for DB2 Data Links Manager	GC09-4829	db2z6x81

Tutorial information

Tutorial information introduces DB2 features and teaches how to perform various tasks.

Table 14. Tutorial information

Name	Form number	PDF file name
Business Intelligence Tutorial: Introduction to the Data Warehouse	No form number	db2tux81
Business Intelligence Tutorial: Extended Lessons in Data Warehousing	No form number	db2tax81
Information Catalog Center Tutorial	No form number	db2aix81
Video Central for e-business Tutorial	No form number	db2twx81
Visual Explain Tutorial	No form number	db2tvx81

Optional component information

The information in this category describes how to work with optional DB2 components.

Table 15. Optional component information

Name	Form number	PDF file name
IBM DB2 Cube Views Guide and Reference	SC18–7298	db2aax81
IBM DB2 Query Patroller Guide: Installation, Administration and Usage Guide	GC09–7658	db2dwx81
IBM DB2 Spatial Extender and Geodetic Extender User's Guide and Reference	SC27-1226	db2sbx81
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x82
DB2 Net Search Extender Administration and User's Guide **Note:** HTML for this document is *not* installed from the HTML documentation CD.	SH12-6740	N/A

Release notes

The release notes provide additional information specific to your product's release and FixPak level. The release notes also provide summaries of the documentation updates incorporated in each release, update, and FixPak.

Table 16. Release notes

Name	Form number	PDF file name
DB2 Release Notes	See note.	See note.
DB2 Installation Notes	Available on product CD-ROM only.	Not available.

Note: The Release Notes are available in:

- XHTML and Text format, on the product CDs
- PDF format, on the PDF Documentation CD

In addition the portions of the Release Notes that discuss *Known Problems and Workarounds* and *Incompatibilities Between Releases* also appear in the DB2 Information Center.

To view the Release Notes in text format on UNIX-based platforms, see the `Release.Notes` file. This file is located in the `DB2DIR/Readme/%L` directory, where `%L` represents the locale name and `DB2DIR` represents:

- For AIX operating systems: `/usr/opt/db2_08_01`
- For all other UNIX-based operating systems: `/opt/IBM/db2/V8.1`

Related tasks:

- "Printing DB2 books from PDF files" on page 154
- "Ordering printed DB2 books" on page 154
- "Invoking contextual help from a DB2 tool" on page 155

Printing DB2 books from PDF files

You can print DB2 books from the PDF files on the *DB2 PDF Documentation* CD. Using Adobe Acrobat Reader, you can print either the entire book or a specific range of pages.

Prerequisites:

Ensure that you have Adobe Acrobat Reader installed. If you need to install Adobe Acrobat Reader, it is available from the Adobe Web site at www.adobe.com

Procedure:

To print a DB2 book from a PDF file:

1. Insert the *DB2 PDF Documentation* CD. On UNIX operating systems, mount the DB2 PDF Documentation CD. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Open index.htm. The file opens in a browser window.
3. Click on the title of the PDF you want to see. The PDF will open in Acrobat Reader.
4. Select **File → Print** to print any portions of the book that you want.

Related concepts:

- "DB2 Information Center" on page 139

Related tasks:

- "Ordering printed DB2 books" on page 154

Related reference:

- "DB2 PDF and printed documentation" on page 149

Ordering printed DB2 books

If you prefer to use hardcopy books, you can order them in one of three ways.

Procedure:

Printed books can be ordered in some countries or regions. Check the IBM Publications website for your country or region to see if this service is available in your country or region. When the publications are available for ordering, you can:

- Contact your IBM authorized dealer or marketing representative. To find a local IBM representative, check the IBM Worldwide Directory of Contacts at www.ibm.com/planetwide
- Phone `1-800-879-2755` in the United States or `1-800-IBM-4YOU` in Canada.

- Visit the IBM Publications Center at http://www.ibm.com/shop/publications/order. The ability to order books from the IBM Publications Center may not be available in all countries.

At the time the DB2 product becomes available, the printed books are the same as those that are available in PDF format on the *DB2 PDF Documentation CD*. Content in the printed books that appears in the *DB2 Information Center CD* is also the same. However, there is some additional content available in DB2 Information Center CD that does not appear anywhere in the PDF books (for example, SQL Administration routines and HTML samples). Not all books available on the DB2 PDF Documentation CD are available for ordering in hardcopy.

Note: The DB2 Information Center is updated more frequently than either the PDF or the hardcopy books; install documentation updates as they become available or refer to the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/ to get the most current information.

Related tasks:

- "Printing DB2 books from PDF files" on page 154

Related reference:

- "DB2 PDF and printed documentation" on page 149

Invoking contextual help from a DB2 tool

Contextual help provides information about the tasks or controls that are associated with a particular window, notebook, wizard, or advisor. Contextual help is available from DB2 administration and development tools that have graphical user interfaces. There are two types of contextual help:

- Help accessed through the **Help** button that is located on each window or notebook
- Infopops, which are pop-up information windows displayed when the mouse cursor is placed over a field or control, or when a field or control is selected in a window, notebook, wizard, or advisor and F1 is pressed.

The **Help** button gives you access to overview, prerequisite, and task information. The infopops describe the individual fields and controls.

Procedure:

To invoke contextual help:

- For window and notebook help, start one of the DB2 tools, then open any window or notebook. Click the **Help** button at the bottom right corner of the window or notebook to invoke the contextual help.

  You can also access the contextual help from the **Help** menu item at the top of each of the DB2 tools centers.

  Within wizards and advisors, click on the Task Overview link on the first page to view contextual help.
- For infopop help about individual controls on a window or notebook, click the control, then click **F1**. Pop-up information containing details about the control is displayed in a yellow window.

Note: To display infopops simply by holding the mouse cursor over a field or control, select the **Automatically display infopops** check box on the **Documentation** page of the Tool Settings notebook.

Similar to infopops, diagnosis pop-up information is another form of context-sensitive help; they contain data entry rules. Diagnosis pop-up information is displayed in a purple window that appears when data that is not valid or that is insufficient is entered. Diagnosis pop-up information can appear for:

– Compulsory fields.
– Fields whose data follows a precise format, such as a date field.

Related tasks:

- "Invoking the DB2 Information Center" on page 147
- "Invoking message help from the command line processor" on page 156
- "Invoking command help from the command line processor" on page 156
- "Invoking SQL state help from the command line processor" on page 157

Invoking message help from the command line processor

Message help describes the cause of a message and describes any action you should take in response to the error.

Procedure:

To invoke message help, open the command line processor and enter:

```
? XXXnnnnn
```

where *XXXnnnnn* represents a valid message identifier.

For example, `? SQL30081` displays help about the SQL30081 message.

Related tasks:

- "Invoking contextual help from a DB2 tool" on page 155
- "Invoking the DB2 Information Center" on page 147
- "Invoking command help from the command line processor" on page 156
- "Invoking SQL state help from the command line processor" on page 157

Invoking command help from the command line processor

Command help explains the syntax of commands in the command line processor.

Procedure:

To invoke command help, open the command line processor and enter:

```
? command
```

where *command* represents a keyword or the entire command.

For example, `? catalog` displays help for all of the CATALOG commands, while `? catalog database` displays help only for the CATALOG DATABASE command.

Related tasks:

- "Invoking contextual help from a DB2 tool" on page 155
- "Invoking the DB2 Information Center" on page 147
- "Invoking message help from the command line processor" on page 156
- "Invoking SQL state help from the command line processor" on page 157

Invoking SQL state help from the command line processor

DB2 Univerrsal Database returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

Procedure:

To invoke SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, `? 08003` displays help for the 08003 SQL state, and `? 08` displays help for the 08 class code.

Related tasks:

- "Invoking the DB2 Information Center" on page 147
- "Invoking message help from the command line processor" on page 156
- "Invoking command help from the command line processor" on page 156

DB2 tutorials

The DB2® tutorials help you learn about various aspects of DB2 Universal Database. The tutorials provide lessons with step-by-step instructions in the areas of developing applications, tuning SQL query performance, working with data warehouses, managing metadata, and developing Web services using DB2.

Before you begin:

You can view the XHTML versions of the tutorials from the Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Some tutorial lessons use sample data or code. See each tutorial for a description of any prerequisites for its specific tasks.

DB2 Universal Database tutorials:

Click on a tutorial title in the following list to view that tutorial.

Business Intelligence Tutorial: Introduction to the Data Warehouse Center
Perform introductory data warehousing tasks using the Data Warehouse Center.

Business Intelligence Tutorial: Extended Lessons in Data Warehousing
Perform advanced data warehousing tasks using the Data Warehouse Center.

Information Catalog Center Tutorial
Create and manage an information catalog to locate and use metadata using the Information Catalog Center.

Visual Explain Tutorial
Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2® products.

DB2 documentation
Troubleshooting information can be found throughout the DB2 Information Center, as well as throughout the PDF books that make up the DB2 library. You can refer to the "Support and troubleshooting" branch of the DB2 Information Center navigation tree (in the left pane of your browser window) to see a complete listing of the DB2 troubleshooting documentation.

DB2 Technical Support Web site
Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs), FixPaks and the latest listing of internal DB2 error codes, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at
http://www.ibm.com/software/data/db2/udb/winos2unix/support

DB2 Problem Determination Tutorial Series
Refer to the DB2 Problem Determination Tutorial Series Web site to find information on how to quickly identify and resolve problems you might encounter while working with DB2 products. One tutorial introduces you to the DB2 problem determination facilities and tools available, and helps you decide when to use them. Other tutorials deal with related topics, such as "Database Engine Problem Determination", "Performance Problem Determination", and "Application Problem Determination".

See the full set of DB2 problem determination tutorials on the DB2 Technical Support site at
http://www.ibm.com/software/data/support/pdm/db2tutorials.html

Related concepts:

- "DB2 Information Center" on page 139

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following list specifies the major accessibility features in DB2® Version 8 products:

- All DB2 functionality is available using the keyboard for navigation instead of the mouse. For more information, see "Keyboard input and navigation" on page 159.

- You can customize the size and color of the fonts on DB2 user interfaces. For more information, see "Accessible display."
- DB2 products support accessibility applications that use the Java™ Accessibility API. For more information, see "Compatibility with assistive technologies."
- DB2 documentation is provided in an accessible format. For more information, see "Accessible documentation."

Keyboard input and navigation

Keyboard input

You can operate the DB2 tools using only the keyboard. You can use keys or key combinations to perform operations that can also be done using a mouse. Standard operating system keystrokes are used for standard operating system operations.

For more information about using keys or key combinations to perform operations, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard navigation

You can navigate the DB2 tools user interface using keys or key combinations.

For more information about using keys or key combinations to navigate the DB2 Tools, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard focus

In UNIX® operating systems, the area of the active window where your keystrokes will have an effect is highlighted.

Accessible display

The DB2 tools have features that improve accessibility for users with low vision or other visual impairments. These accessibility enhancements include support for customizable font properties.

Font settings

You can select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

For more information about specifying font settings, see Changing the fonts for menus and text: Common GUI help.

Non-dependence on color

You do not need to distinguish between colors in order to use any of the functions in this product.

Compatibility with assistive technologies

The DB2 tools interfaces support the Java Accessibility API, which enables you to use screen readers and other assistive technologies with DB2 products.

Accessible documentation

Documentation for DB2 is provided in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen-reader.

Related concepts:

• "Dotted decimal syntax diagrams" on page 160

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the Information Center using a screen reader.

In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.
- * means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Notes:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.
- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Related reference:

- "How to read the syntax diagrams" in the *SQL Reference, Volume 2*

Common Criteria certification of DB2 Universal Database products

DB2 Universal Database is being evaluated for certification under the Common Criteria at evaluation assurance level 4 (EAL4). For more information about Common Criteria, see the Common Criteria web site at: http://niap.nist.gov/cc-scheme/.

Appendix B. Notices

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

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

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

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

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

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

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

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

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

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

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information 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.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Information Integrator
DB2 Query Patroller
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA
iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Index

D

E

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at http://www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at http://www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, product downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide

Part Number: SDB2-CONN-SU

Printed in USA

Spine information:

IBM®

Connectivity Supplement

Version 8