Installation and Administration Guide

Mainframe Connect DB2 UDB Option

12.6

[IBM CICS]

DOCUMENT ID: DC36525-01-1260-01

LAST REVISED: May 2005

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

This book describes how to install and configure the DB2 UDB Option for CICS. It includes planning considerations, installation instructions, configuration instructions, and troubleshooting information.

This preface includes the following topics:

Audience

These specific guidelines and step-by-step instructions are intended for those who install, configure, and maintain Sybase mainframe client/server components on an IBM z/Series mainframe. This guide is for anyone who performs tasks, such as a:

- CICS system programmer
- DB2 database administrator
- DirectConnect administrator

This guide assumes you have a working knowledge of system administration for your environment. For a z/OS-CICS environment, for example, you should be familiar with z/OS and CICS system administrative procedures, such as defining CICS resources and SNA definitions.

Product name changes

The following table describes new names for products in the 12.6 release of the Mainframe Connect Integrated Product Set (IPS).

Old product names	New product name
• Open ClientConnect™ for CICS • Open ClientCONNECT for CICS	Mainframe Connect Client Option for CICS
• Open Client Connect for IMS and MVS • Open ClientCONNECT for IMS and MVS	Mainframe Connect Client Option for IMS and MVS
• Open ServerConnect™ for CICS • Open ServerCONNECT for CICS	Mainframe Connect Server Option for CICS
• Open ServerConnect for IMS and MVS • Open ServerCONNECT for IMS and MVS	Mainframe Connect Server Option for IMS and MVS
• MainframeConnect™ for DB2 UDB • MainframeCONNECT for DB2/MVS-CICS	Mainframe Connect DB2 UDB Option for CICS
• DirectConnect™ for OS/390 • DirectCONNECT for DB2/MVS	DirectConnect for z/OS

How to use this book

This guide consists of five sections: planning, installing, configuring, troubleshooting, and appendix. The first four sections contain chapters that describe sets of tasks. The last section contains useful reference information.

The DB2 UDB Option for CICS requires the Server Option for CICS to be installed. Therefore, the Server Option for CICS is shipped with the DB2 UDB Option for CICS. You will find instructions for installing the Server Option for CICS in the Mainframe Connect Server Option for CICS *Installation and Administration Guide*, which is also shipped with the DB2 UDB Option for CICS.

The following table shows tasks you perform using this guide and the Mainframe Connect Server Option for CICS *Installation and Administration Guide*.

To		See
Understand	DB2 UDB Option for CICS components	Chapter 1, “Understanding the DB2 UDB Option for CICS”
Plan	Installation and configuration	Chapter 2, “Pre-installation Tasks”
Install	DB2 UDB Option for CICS	• Mainframe Connect Server Option for CICS *Installation and Administration Guide* • Chapter 3, “Installation”

To		See
Configure	DB2 UDB Option for CICS	• Chapter 4, "Configuring the DB2 UDB Option for CICS" • Chapter 5, "Customizing the DB2 UDB Option for CICS Messages"
Troubleshoot	Problems with client access to data on the mainframe	Chapter 8, "Troubleshooting"
Reference	• DB2 UDB Option for CICS-specific messages • Security information • Default translation tables to customize SBCSs • Unfamiliar terms and acronyms	• Appendix A, "Error Messages" • Mainframe Connect Server Option for CICS *Installation and Administration Guide* • Glossary

Related documents

In addition to this book, you also need the Mainframe Connect Server Option for CICS *Installation and Administration Guide*, which ships with the DB2 UDB Option for CICS.

For the latest product information, refer to the release bulletins for the DB2 UDB Option for CICS and the Server Option for CICS.

You may also need to refer to the following documentation:

- Mainframe Connect DirectConnect for z/OS Option *Installation Guide*
- Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*
- Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*
- Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide* for DirectConnect
- Mainframe Connect Server Option *Programmer's Reference for Remote Stored Procedures*
- Mainframe Connect Client Option *Programmer's Reference for Client Services Applications*
- Mainframe Connect Client Option and Server Option *Messages and Codes Guide*

Other sources of information

Use the Sybase® Getting Started CD, the SyBooks™ CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

Syntax statements that display options for a command look like this:

```
COMMAND [object_name, [ {TRUE | FALSE} ] ]
```

The following table explains the syntax conventions used in this guide.

Table 1: Syntax conventions

Symbol	
`( )`	Parentheses indicate that they are included as part of the command.
`{ }`	Braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you type the option.
`[ ]`	Brackets indicate that you can choose either one or more of the enclosed options, or none of the options. Do not type the brackets when you type the options.
`|`	The vertical bar indicates that you can select only one of the options shown. Do not type the bar in your command.
`,`	The comma indicates that you can choose one or more of the options shown. Separate each choice by using a comma as part of the command.

The following style conventions are used in this book:

Table 2: Style conventions

This type of information	Looks like this
Directories and files	*econnect\ServerName\cfg*
Programs, utilities, procedures, and commands	snrfck
Properties	Allocate
Options	connect
Code examples and text on screen	`this font`
Commands you should enter	*this font*
Variables you should replace with the appropriate value for your system when entering a command	*`this font`*

All other names and terms are in regular typeface.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1

Understanding the DB2 UDB Option for CICS

The DB2 UDB Option for CICS is part of the Sybase Mainframe Connect Integrated Product Set (IPS). For an explanation of how the DB2 UDB Option for CICS works with other Mainframe Connect products, see the Mainframe Connect IPS *Overview Guide*.

The DB2 UDB Option for CICS is a Server Option application providing client applications with access to DB2 and nonrelational data sources on an IBM mainframe running z/OS and CICS.

This chapter provides an overview of the DB2 UDB Option for CICS and contains the following sections:

Related products

The DB2 UDB Option for CICS primarily operates with two other Sybase products: DirectConnect for z/OS Option and the Server Option.

DirectConnect for z/OS Option

The DB2 UDB Option for CICS operates in both a two-tier, gateway-less environment and a three-tier, gateway-enabled environment. The Sybase gateway component is DirectConnect, which provides the communication link between a mainframe and a remote network of clients.

For more information on the DirectConnect for z/OS Option, see the documentation for that product.

Server Option

The Server Option uses Open Server architecture to allow the mainframe to act as a server. The Server Option is available for CICS, IMS, and native z/OS. The DB2 UDB Option for CICS runs on top of the Server Option for CICS. Together, they support the features described in this chapter.

The DB2 UDB Option for CICS uses the Server Option connectivity for multiple protocols and for support and customization of character sets.

Multiple connectivity protocols

Using the Server Option, the DB2 UDB Option can achieve connectivity with both of the following connectivity protocols:

- SNA LU 6.2 connectivity
- TCP/IP connectivity

Character set support and customization

The DB2 UDB Option for CICS (with the Server Option) supports the following character sets:

- ascii_8, iso-1
- cp-850
- cp-437, eucjis
- SJIS
- dec-kanji
- roman8 (for HP)
- mac
- cp819 (UNIX ascii iso_1)
- cp500 (mainframe ebcdic iso_1)
- utf8

In addition, messages are provided in multiple languages, such as U.S. English, French, or German.

DB2 UDB Option features

The DB2 UDB Option for CICS supports the following features:

- Translation between DB2 and Sybase datatypes and error codes.
- DB2 catalog queries.
- 31-digit decimals; the DB2 UDB Option for CICS supports the DB2 extension to its decimal datatype, *DECIMAL*(31,31).
- *LONG VARCHARs* (literal strings greater than 254 bytes) can be embedded in an insert or update language statement.

In addition, the DB2 UDB Option for CICS supports:

- Long-running transactions
- Character set customization
- Access to relational and nonrelational data
- Dynamic SQL DB2 data access
- User-defined datatypes
- Exit routines

Long-running transactions

The DB2 UDB Option for CICS supports user-defined, long-running transactions. Because transactions can remain active after all results are sent, multiple requests can participate in the same unit of work. Therefore, you have more control over individual or group commits or rollbacks on mainframe databases.

Character set customization

The Server Option for CICS provides character sets that you can customize if you want to change languages. You can create up to 99 new character sets by modifying the shipped base tables.

Access to relational and nonrelational data

In addition to relational data support, the DB2 UDB Option for CICS provides access to nonrelational data through RSPs and RPCs. (RSPs require the Server Option API.) For more information, see the Mainframe Connect Server Option *Programmer's Reference* for the appropriate language. RSP, PL/1, and COBOL versions of this guide are available.

Dynamic SQL DB2 data access

The DB2 UDB Option for CICS supports the following:

- Cursor and dynamic events
- Updates and inquiries against DB2 data using DB2 dynamic SQL
- Dynamic access to DB2 data and CICS programs

User-defined datatypes

Client applications can process user datatypes that return descriptive information about DB2 columns.

Exit routines

The DB2 UDB Option for CICS provides request, result, and parse exit routines.

Components and functionality

This section lists the components of the DB2 UDB Option for CICS and the applications with which the DB2 UDB Option for CICS communicates:

- Components of the DB2 UDB Option for CICS
- Communication with other applications

Components of the DB2 UDB Option for CICS

The DB2 UDB Option for CICS consists of the transaction AMD2 and uses the host request library DB2 table, both of which reside on the mainframe.

Transaction processor AMD2

AMD2 is the default name of the CICS transaction in the DB2 UDB Option for CICS that allows clients to submit SQL statements to DB2 or RSP requests.

Note A separate installed CICS region with AMD2 is required for each DB2 subsystem accessed.

DirectConnect invokes AMD2 when it sends a client SQL request to the mainframe. AMD2 then submits the SQL request to DB2, using the DB2 dynamic SQL facility.

AMD2 returns DB2 results and messages to DirectConnect and then terminates. However, in the case of long transactions or allocation on connect, AMD2 terminates only after the client explicitly ends the transaction.

Host request library DB2 table

The host request library is a DB2 table used for centrally storing SQL statements. This library makes the requests available to all DirectConnect users.

The DB2 UDB Option for CICS retrieves the specified SQL request from the host request library and submits the request to DB2 for processing. The DB2 UDB Option for CICS returns the results of the request to the client application.

Communication with other applications

The DB2 UDB Option for CICS and DirectConnect handle the communications between the client application, DB2, and other products.

The products discussed in the following sections comprise the major parts of the communication path from a client to DB2. They affect either the processing or the content of client requests.

DirectConnect for z/OS Option

Client application requests use DirectConnect to access the DB2 UDB Option for CICS on the mainframe. Client application requests are initiated by Sybase and third-party client applications.

To establish connectivity between DirectConnect and the DB2 UDB Option for CICS, use SNA LU 6.2 or TCP/IP connectivity and the Tabular Data Stream™ (TDS) protocol. See the Mainframe Connect DirectConnect for z/OS Option *Installation Guide* for more information.

Client applications

Client applications send requests to DirectConnect (or to other servers for forwarding to DirectConnect). Client requests can be sent through ODBC, DB-Library™, CT-Library™, or Open Client™ applications, such as isql, Data Workbench®, APT Workbench™, and front-end products from independent software vendors.

DirectConnect provides the link between the DB2 UDB Option for CICS and the clients. Therefore, after the DB2 UDB Option for CICS is installed and configured, the clients do not need to consider communication issues with the DB2 UDB Option for CICS.

Other mainframe applications

The DB2 UDB Option for CICS interacts with these mainframe products:

- DB2
- CICS
- Resource Access Control Facility (RACF), Advanced Communications Function 2 (ACF2), or TopSecret

Other servers

Adaptive Server® Enterprise/Component Integration Services (ASE/CIS) and Replication Server® are separate Sybase products that work with DirectConnect and can be optional parts of the DB2 UDB Option for CICS communication path.

DB2 UDB Option communication path for three-tier environments

For client access to DB2 data, the DB2 UDB Option for CICS must be installed on the mainframe, and DirectConnect must be installed on a LAN server. The DB2 UDB Option for CICS and DirectConnect work together to provide DB2 data to clients as follows (see Figure 1-1):

1 A client sends SQL requests to DirectConnect.

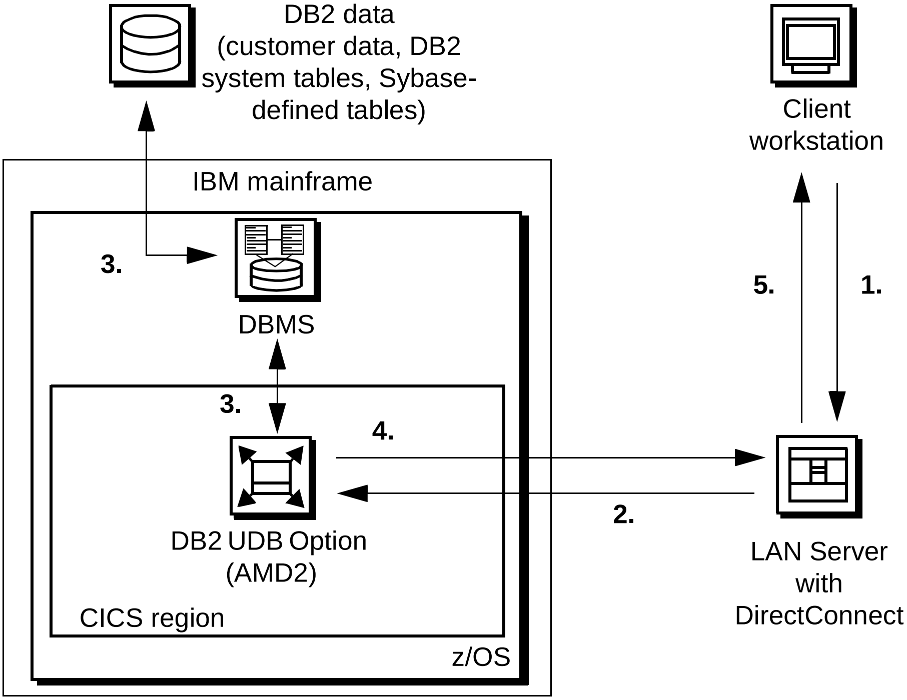
2 DirectConnect forwards the requests to the CICS transaction (AMD2) on the mainframe (using SNA or TCP/IP connectivity on the LAN server).

3 AMD2 processes the client SQL requests using the dynamic SQL capabilities of DB2.

4 AMD2 returns the DB2 result sets and messages to DirectConnect (using SNA or TCP/IP connectivity on the LAN server).

5 DirectConnect returns the results and messages to the client.

Figure 1-1: DB2 UDB Option communication path for three-tier



DB2 UDB Option communication path for two-tier environments

For gateway-less client access to DB2 data, the DB2 UDB Option must be installed on the mainframe. The DB2 UDB Option provides DB2 data to clients as follows (see Figure 1-2):

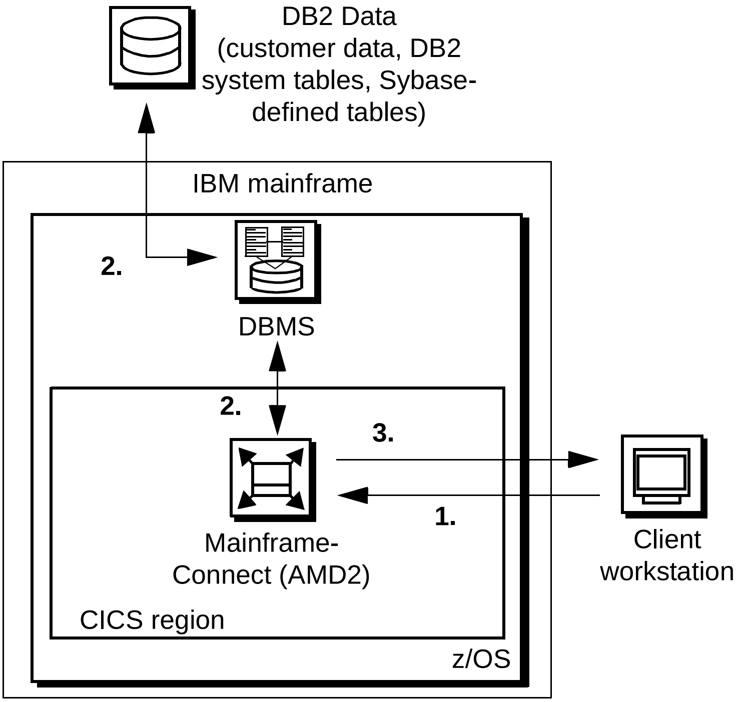
1 A client sends SQL requests to the CICS transaction (AMD2) on the mainframe (using TCP/IP connectivity).

2 AMD2 processes the client SQL requests using the dynamic SQL capabilities of DB2.

3 AMD2 returns the DB2 result sets and messages to the client (using TCP/IP connectivity).

See the following figure for an illustration of this process.

Figure 1-2: DB2 UDB Option communication path for two-tier gateway-less



CHAPTER 2 Pre-installation Tasks

This chapter explains issues to consider before installing the DB2 UDB Option for CICS and covers the following topics:

Planning the installation

This section includes the following topics:

- Installation media
- What to install
- Pre-installation tasks
- Task list

Installation media

The Server Option is now distributed on CD or in downloadable form. The Server Option is no longer distributed on tape.

Note EBFs for the Server Option are no longer distributed by tape. For information on obtaining the latest EBFs for the Server Option, see the Mainframe Connect Server Option for CICS *Release Bulletin*.

What to install

To use the full functionality of the DB2 UDB Option for CICS, install the following:

- The Server Option for CICS, which contains files for LU 6.2 and TCP/IP environment connections
- The DB2 UDB Option for CICS, which contains the Server Option modules needed for the DB2 UDB Option for CICS to communicate with LAN clients

Note For details specific to installing and configuring the Server Option for CICS, see the Mainframe Connect Server Option for CICS *Installation and Administration Guide*.

Pre-installation tasks

Installation requires completing the following pre-installation tasks, which are explained in the following subsections. You should skip those tasks that do not pertain to the option or options you have chosen to install.

1. Verify mainframe access components
2. Determine JCL and system information
3. Determine CICS and DB2 UDB information
4. Determine compiler information
5. Determine Server Option information
6. Determine ftp information
7. Plan the security requirements
8. Identify the change control requirements
9. Back up the release libraries (upgrades only)
10. Determine the library names

11. Verify the connectivity

Note To determine whether any new release levels or bug fixes of vendor or Sybase products are required, consult the Mainframe Connect DB2 UDB Option for CICS *Release Bulletin* for the most current information.

Task list

Following is the list of tasks to be performed prior to installation.

1. Verify mainframe access components

Verify that you have the compatible mainframe access components shown in the following table before you install the DB2 UDB Option for CICS.

Table 2-1: Mainframe access components

Component	Version
Server Option for CICS	12.6
DirectConnect for z/OS Option	12.6

2. Determine JCL and system information

Determine the following information to be used in the installation procedure:

- JCL jobcard values, which are used in the final installation jobs run in TSO.
- High-level qualifier, which is used as a prefix for data sets generated during installation.
- Volume serial number, which indicates where generated data sets are cataloged.
- Unit parameter value, which indicates the device requirements for cataloging generated data sets.
- Work unit, which is for the use of temporary work data sets.
- Customer CICS, IMS, and MVS LOADLIBs, which are pre-cataloged partitioned data sets (PDSs) or partitioned data sets extended (PDSE) into which configuration modules and sample programs are to be linked.

3. Determine CICS and DB2 UDB information

Determine the following information:

- High-level qualifier for CICS system data sets.
- RDO data set name (DSN), which is the name of the CICS RDO (DFHCSD) containing the application resource definitions used by your CICS region.
- RDO group list, which is the RDO group list used by your CICS region when executing an initial start.
- The CICS region APPLID, which is the VTAM APPLID for your CICS region.
- DB2 system data sets high-level qualifier, which is the high-level qualifier used for DB2 system data sets.
- DB2 exit data set name (DSN).
- DB2 DSN.

4. Determine compiler information

Determine the following information if you intend to install an API component:

- LE370 high-level qualifier, which is used for the Language Environment 370.
- COBOL compiler name, which is the module used to execute COBOL in your environment.
- COBOL compiler LOADLIB, which is the system LOADLIB where your COBOL compiler module resides.
- PL/1 compiler name, which is the module used to execute PL/1 in your environment.
- PL/1 compiler LOADLIB, which is the system LOADLIB where your PL/1 compiler module resides.
- C compiler data sets high-level qualifier, which is the high-level qualifier used for C.
- TCP/IP data sets high-level qualifier.

5. Determine Server Option information

Determine the following information for use in installing the Server Option:

- TCP address space name.
- Remote server name, which is the name by which your Server Option applications will refer to the remote server.
- Remote TCP host name, which is the DNS name for the remote server.
- Remote server TCP host port, which is the TCP/IP port used by the remote server.

6. Determine ftp information

Determine the following information needed to establish an ftp connection to your mainframe:

- User ID.
- Password.
- Mainframe host name.
- Control port number, which is the listener port used by your mainframe ftp server, usually 21.
- TCP address space name.
- Volume serial number or unit. You may specify either a volume serial number (VOL=SER) and unit assignment for ftp to use or allow ftp to use default values.
- Log path name, which indicates where ftp log information is to be written.

7. Plan the security requirements

Review your security requirements with your security administrator. You may also need to consult with your network administrator.

8. Identify the change control requirements

Create a change control plan that includes:

- All the tasks that need to be considered for installation

- The different groups that need to be aware of the environment change, for example, field personnel and groups involved in administering applications, z/OS, security, change control, and scheduling
- A schedule, including cut-off dates for specific tasks

9. Back up the release libraries (upgrades only)

If you are upgrading an existing release, Sybase strongly recommends that you back up the entire set of release libraries before beginning this installation. Also, see “Migrating from previous installations” for more information on upgrading.

10. Determine the library names

The shipped library names are unique for this release. If you are upgrading, decide whether you want to use your current library names. If this is a new release, you still might want to consider how to name the files.

You do not have to remove previous versions from your Sybase libraries because default names shipped with this release create an entirely unique set of release libraries. You can change them, however, based on naming standards at your site.

Note When the upgrade is complete and tested, be sure to replace the old LOADLIB name or add the new LOADLIB name to the DFHRPL concatenation for the selected CICS region(s), as described in the installation instructions.

If you are going to use the old Sybase library names, delete all members before installing the new ones with the new version.

11. Verify the connectivity

Use the standard LAN ping utility to ensure connectivity between z/OS and the workstation running Adaptive Server® Enterprise (ASE).

Migrating from previous installations

This section covers issues to consider when upgrading from a previous installation of the DB2 UDB Option for CICS.

New RDO and DB2 plans

RDO entries for DB2 have changed from the previous release. Be sure to run the install jobs for updating RDO and DB2 plans. TRS CSP entries are now combined with the DB2 entries.

Preserving customized messages on AMD2CAMX

Customized messages are a part of the AMD2 message module, AMD2CAMX. Previous mainframe access product releases supplied messages through DB2 Version 8.1. Depending on the DB2 version at your site, consider the following:

- If your site has *not* added customized messages (other than updating those supplied as standard for DB2 Version 8.1), you can install the DB2 UDB Option for CICS Version 12.6 without first preserving your customized messages.

  Note The DB2 UDB Option for CICS message table includes messages supplied through DB2 Version 8.1. Customized messages are always added to the messages supplied by the DB2 UDB Option for CICS; they usually do not replace them.

- If your site added customized messages that are not DB2 standard messages, preserve them now if you do not want the installation process to overwrite them. You can then add your customized messages to the AMD2CAMX message table after installing this release. For more information, see “Add entries for new SQLCODEs” in Chapter 5, “Customizing the DB2 UDB Option for CICS Messages.”

After you complete and test the upgrade, replace the old LOADLIB name with the new one in the DFHRPL concatenation, which is in the CICS start-up procedure for that CICS region. These installation steps are described in Chapter 3, “Installation.”

CHAPTER 3

Installation

This chapter describes the following topics.

Topic	Page
Installation	17
Post-installation steps	24
Libraries and samples	25

Before you begin

Be sure you completed the tasks in Chapter 2, “Pre-installation Tasks.”

Installation

The following two procedures describe the installation steps necessary to install all Mainframe Connect options from the InstallShield wizard and to complete the installation for the Server Option for CICS. You should skip those installation steps that do not pertain to the option or options you have chosen to install.

Note The InstallShield wizard runs only on Windows.

❖ Installing from the InstallShield wizard

1 Start the InstallShield wizard from CD by executing *setupwin.exe,* which is in the root directory.

The initial dialog box displays the options available for installation. Click Next and Back to navigate through the wizard. To cancel the installation, click Cancel.

2 Click Next, and accept the terms of the user-license agreement by selecting your country in the drop-down list and selecting the option to indicate that you agree with the terms.

3 Click Next, and select the components you want to install.

Note If you are installing the Server Option for CICS API or the DB2 UDB Option for CICS, the Server Option for CICS Runtime component will be automatically selected as you proceed to the next screen.

4 Enter the license keys for the components you purchased.

5 Click Next, and provide the following JCL and system information:

- *JCL Line 1-3*: Enter a valid jobcard. This is used to run the final installation jobs in TSO.
- *High Level Qualifier*: The high-level qualifier is used as a prefix for all data sets generated during installation.
- *Volume*: The volume serial number indicates where generated data sets are cataloged.
- *Unit*: The unit parameter value indicates the device requirements for cataloging generated data sets.
- *Work Unit*: This is for the use of temporary work data sets.
- *Customer CICS, IMS, and MVS Loadlibs*: These are pre-cataloged partitioned data sets (PDSs) or partitioned data sets extended (PDSE) into which configuration modules and sample programs are to be linked. For CICS, this data set should be in the DFHRPL configuration ahead of other Sybase libraries.

Click Next.

6 If you have chosen to install an option that uses CICS, DB2, or IMS, provide the following information where it applies. Otherwise, skip to the next step.

- *CICS system datasets hlq*: The high-level qualifier for CICS system data sets is used to locate *SDFHLOAD* and other CICS libraries.
- *RDO Dataset*: The RDO data set name is the name of the CICS RDO (DFHCSD) containing the application resource definitions used by your CICS region.
- *RDO Group List*: The RDO group list is the RDO group list used by your CICS region when executing an initial start.
- *CICS Region Applid*: The CICS region APPLID is the VTAM APPLID for your CICS region.

- *DB2 system datasets hlq*: The DB2 system data sets high-level qualifier is used for DB2 system data sets.
- *DB2 Exit Dataset*: This is the name of the DB2 exit data set used by your DB2 region.
- *DB2 DSN Name*: This is the data set name (DSN) of your DB2 region.
- *IMS datasets hlq*: The high-level qualifier for IMS system data sets is used to locate IMS libraries.

Click Next.

7 If you have chosen to install an API component, provide the following compiler information, which is used to configure JCL for compiling sample programs. Otherwise, skip to the next step.

- *LE/370 datasets hlq*: The LE370 high-level qualifier is used for the Language Environment 370 and is used here to locate data sets like *CEELKED*.
- *COBOL Compiler Name*: The COBOL compiler name is the module used to execute COBOL in your environment.
- *COBOL Compiler Loadlib*: The COBOL compiler LOADLIB is the system LOADLIB in which your COBOL compiler module resides.
- *PLI Compiler Name*: The PLI compiler name is the module used to execute PLI in your environment.
- *PLI Compiler Loadlib*: The PLI compiler LOADLIB is the system LOADLIB in which your PLI compiler module resides.
- *C compiler datasets hlq*: The C compiler data sets high-level qualifier is the high-level qualifier used for C and is used to locate data sets like *SBCCMP*.
- *TCP/IP datasets hlq*: The TCP/IP data sets high-level qualifier is used to locate data sets like *SEZATCP*.

Click Next.

8 If you have chosen to install the Client Option for CICS, provide the following information for configuring a host connection definition for the Client Option. Otherwise, skip to the next step.

- *TCP Address Space Name*: This is the name of your TCP/IP region.
- *Server Name*: This is the name by which your Client Option applications refers to the remote server.

- *Server TCP Host Name*: This is the DNS name for the remote server.
- *Server TCP Host Port*: This is the TCP/IP port used by the remote server.

Click Next.

9 If you have chosen to install the Server Option for CICS or the DB2 UDB Option for CICS, provide the following information for configuring a TCP/IP listener for these options. Otherwise, skip to the next step.

- *TCP Address Space Name*: This is the name of your TCP/IP region.
- *Listener Port*: This is the port on which the option listens.

Note The Server Option for CICS and the DB2 UDB Option for CICS share the same TCP/IP listener.

Click Next.

10 Click Next until the wizard displays the information you entered in steps 5 through 8. Review this information and, if necessary, click Back to return to previous screens and make corrections.

11 Click Next until the wizard displays a dialog box for ftp information. Provide the following data for establishing an ftp session to your mainframe:

- *Userid*: This is the mainframe user ID for the ftp session.
- *Password*: This is the password for the ftp session.
- *Mainframe Host Name*: This is the mainframe DNS name.
- *FTP Port*: This is the control port used by your mainframe ftp server, usually 21.
- *VOL/UNIT Assignment*: Specify either a volume serial number and unit assignment for ftp, or allow ftp to use default values.

Note If you specify a volume serial number that does not exist, ftp hangs until the mainframe responds to a message requesting that the volume be mounted.

- *Log FTP Commands*: This indicates where ftp log information is to be written. This log information may be useful in troubleshooting ftp problems.

The InstallShield wizard will create JCL and upload the selected components to your mainframe once you click Next.

12 Close the InstallShield wizard.

To complete the installation of your Mainframe Connect components, review and submit JCL from TSO. If you are installing multiple components, Sybase strongly suggests you install in the following sequence:

1. Client Option for CICS
2. Server Option for CICS
3. DB2 UDB Option for CICS
4. Any other options

Use the following procedure to complete your installation.

❖ Completing the installation

1. Locate the installation JCL for the DB2 UDB Option for CICS in *hlq.AMD2126.CICSDB2.JCL*, where *hlq* is the high-level qualifier you specified in step 5.
2. Run the following jobs in the order they are described here, where *x* is an integer that denotes the order in which the job is to be run in the overall sequence of jobs. Ignore jobs that are not present or relevant to the option you are installing.

   - *IxRECV*: This job runs IKJEFT01 to use the TSO RECEIVE command to build and populate the product libraries.
   - *IxRDO*: This job runs the CICS Resource Definition Utility, DFHCSDUP, to define the transaction, program, and file entries for the DB2 UDB Option for CICS. If your CICS region has had a previous version of the DB2 UDB Option, you may need to uncomment or change the DELETE and REMOVE entries at the top of the RDO input.
   - *IxVSAM*: This job allocates the VSAM data sets used for error and trace logging.
   - *IxLIC*: This job populates the license data set for the DB2 UDB for CICS. Since this data set is used for all of the Mainframe Connect IPS options, the Create step may fail if the data set has already been defined. This error may be safely ignored.

- *IxRPC*: This job adds definitions to the *SYRPCFIL* dataset, which is created during the installation of the Server Option and required for the DB2 UDB Option. If DB2 is not to be used in this CICS region, run the *IxRPC1* job instead.
- *IxBIND*: This job binds the DB2 plans required for the DB2 UDB Option.

Note

- The asterisk and period (*.) before *AMD2PKG.** in the PKLIST parameter of the *BINDPLAN* job may not be required on your DB2 system.
- In the Server Option for CICS configuration module *SYGWXCPH*, there is a parameter called CHARSETSRV that can be set to "iso_1" or "utf8". The DB2 UDB Option for CICS works with the CHARSETSRV parameter. This parameter determines which of two DB2 packages, *AMD2DB2* or *AMD2DBA*, are used. If the value of CHARSETSRV is "iso_1," the DB2 packages must be bound with ENCODING(EBCDIC), which is the default. If the value of CHARSETSRV is "utf8," the DB2 packages must be bound with ENCODING(1208).

- *IxDELETE*: This optional job deletes the data sets in the TSO XMIT format used for the installation.

3 Run the following jobs if you want to compile and link-edit the sample applications provided with the DB2 UDB for CICS:

- *PAMDCAMX*: This job assembles and links the message module for the DB2 UDB Option.
- *PAMDPXIT*: This job compiles and links the parser exit module.
- *PASMEXIT*: This job assembles and links various exits for the DB2 UDB Option.

4 If you are migrating from a previous version of the DB2 UDB Option for CICS, use the *TRSDROP* member in the *CTRL* library as input to SPUFI to drop the old catalog stored procedure (CSP) tables and views.

5 Create tables and views for the CSPs as input to SPUFI using member *TRSCREAT* or *TRSCREA2* in the *CTRL* library. The following points will help you decide which member to use:

- *TRSCREAT* verifies whether the current user has some privilege on a requested object using the current *sqlid*. If the current *sqlid* does not have permissions on an object, granted either explicitly or to PUBLIC, then the CSPs do not return information for that object.
- *TRSCREA2* enables faster CSP processing because it does not verify that the current user has a privilege on a requested object.

  The following conditions apply when using *TRSCREA2*:

  - Anyone who can execute the CSP can obtain information about all DB2 objects, even if that user does not have privileges on those objects.
  - The accessible_table column in sp_server_info is ineffective.

6 Edit and run the *TRSGRANT* member of *CTRL* using SPUFI to grant authorization for users to run the CSPs.

7 Create a collection ID by issuing the following SQL command to DB2 (using a utility like TSO batch or SPUFI):

```
GRANT PACKADM ON COLLECTION AMD2PKG TO userid
```

8 For each additional DB2 subsystem to be accessed by the DB2 UDB Option for CICS, repeat steps 2 (*IxBIND*), 4, 5, 6, and 7.

9 Edit the CICS start-up procedure to include the DB2 UDB Option for CICS LOADLIB library in the *DFHRPL* concatenation. This library should be placed before the load library for the Server Option for CICS.

10 This step is optional. Alter the message table using the *AMD2CAMX* member of *SOURCE* as a prototype. Assemble and link the message table using the *AMD2CAMX* member of *JCL*. For details on changing the message table, see Chapter 5, “Customizing the DB2 UDB Option for CICS Messages.”

11 Recycle your CICS region.

12 For each additional CICS region in which the DB2 UDB Option is to be installed, repeat steps 2 (*IxRDO*), 9, 10, and 11, making adjustments to the JCL where appropriate.

Post-installation steps

The DB2 The DB2 data administrator should complete the following procedure after installing the DB2 UDB Option for CICS.

❖ **Authorizing and verifying**

1 Everyone who uses the DB2 UDB Option for CICS to access DB2 must be authorized to execute the DB2 application plan on each DB2 system. This is the application plan created that contains all of the DB2 application packages. Privileges can be granted to all users or only to specific DB2 users. To issue the GRANT statement, use either of the following forms:

```
GRANT EXECUTE ON PLAN AMD2PLAN TO userid
GRANT EXECUTE ON PLAN AMD2PLAN TO PUBLIC
```

If the plan name has been changed during installation, change *AMD2PLAN* to the new name here.

2 Users must have EXECUTE privileges granted on the packages created on each DB2 system. Use either of the following forms:

```
GRANT EXECUTE ON PACKAGE AMD2PKG.* TO userid
GRANT EXECUTE ON PACKAGE AMD2PKG.* TO PUBLIC
```

If the collection name has been changed during installation, change *AMD2PKG* to the new name here.

3 To use the CSPs, users must be authorized to select from the following tables on each DB2 system:

- SYSIBM.SYSCOLAUTH
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSDATABASE
- SYSIBM.FOREIGNKEYS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSKEYS
- SYSIBM.SYSRELS
- SYSIBM.SYSSYNONYMS
- SYSIBM.SYSTABAUTH
- SYSIBM.SYSTABLES

Some front-end applications access these system tables directly. See the application documentation for these applications for more information. Also, any users accessing DB2 objects must be granted the appropriate authorizations on those objects.

4 Before configuring the DB2 UDB Option for CICS, you should verify that the link between the DB2 UDB Option for CICS and DB2 is functioning correctly. To do this, sign on to CICS, clear the screen, and enter the following command:

```
AMD2
```

A message similar to the following displays:

```
MainframeConnect DB2 UDB Option for
CICS/12.6.0.0/P/IBM Z/126-011
DB2 test successful
```

If the UDB DB2 Option for CICS cannot access DB2, a message with the DB2 SQLCODE appears. For example:

```
MainframeConnect DB2 UDB Option for
CICS/12.6.0.0/P/IBM Z/126-011
DB2 Error was detected the SQL code is –805
```

See the IBM DB2 manuals for an explanation of the SQL codes.

Libraries and samples

For a list and description of the libraries, sample programs, JCL, and transactions for your product, see the *CONTENTS* member of the *JCL* data set.

CHAPTER 4

Configuring the DB2 UDB Option for CICS

After you install the DB2 UDB Option for CICS, you are ready to configure the DB2 UDB Option for CICS for your site. This chapter summarizes the procedure used to enable communication between the DB2 UDB Option for CICS and DirectConnect. This chapter is divided into the following topics:

Changing the DB2 UDB Option for CICS configuration

Use the DB2 UDB Option Configuration Tran ID Selection panel and the DB2 UDB Option Configuration panels to alter the DB2 UDB Option for CICS configuration settings as necessary. The settings are automatically saved in the configuration dataset.

Note When you change configuration properties in the DB2 UDB Option Configuration panel, the new values remain in effect until they are changed again.

❖ Changing the DB2 UDB Option for CICS configuration settings

1 Using a 3270 terminal emulator, sign on to CICS, clear the panel, and enter the following command:

```
AMCF
```

The DB2 UDB Option Configuration Tran ID Selection panel appears as shown in the following figure. This panel enables you to select the transaction ID for which you want to change the configuration, or it begins the process of configuring a new transaction ID.

Note The first time the previous panel appears, no Profile Name entries appear.

2 Either select the transaction ID from the list and press Enter, or press F4 to add a new transaction ID.

The DB2 UDB Option Configuration panel appears.

3 Change the settings to the values you prefer (as explained in “Understanding the configuration properties”), and press Enter.

When you add a new transaction ID, the Transaction ID field is blank. You must enter a 4-character transaction ID in the Transaction ID field.

Understanding the configuration properties

This section describes the configuration properties you can set in the DB2 UDB Option Configuration panel.

Transaction ID

The *Transaction ID* is the ID that you selected in the DB2 UDB Option for CICS Configuration Tran ID Selection panel in step 1. If you are adding a transaction ID, this field is blank, and you must provide a value. Any additional transactions set up for the DB2 UDB Option for CICS must be defined in RDO.

Request exit name

The *Request exit name* specifies the name of the request validation exit program. This name can contain up to 8 characters. If a request validation exit program does not exist, specify NONE (the default).

For more information, see “Request/Result validation exits.”

Result exit name

The *Result exit name* specifies the name of the result validation exit program. This name can contain up to 8 characters. If a result validation exit program does not exist, specify NONE (the default).

For more information, see “Request/Result validation exits.”

Parse exit name

The *Parse exit name* identifies the CICS program to which AMD2 links as a parse exit routine. This name can contain up to 8 characters. If a parse exit routine does not exist, specify NONE (the default).

For more information, see “Parse exit routine.”

Group ID exit name

The *Group ID exit name* supplies CSPs with the group IDs of which the exit user ID is a member. The CSPs use these Group IDs to obtain all the tables to which the exit user ID has access.

If you are not using the RACF, ACF2, or Top Secret security packages, you can write an exit program to build a list of group IDs as input to CSPs.

If you are using the RACF, ACF2, or Top Secret security packages (see the Mainframe Connect Server Option for CICS *Installation and Administration Guide* for information about security), use one of the supplied exits.

Sybase provides the following Group ID exit names for RACF, Top Secret, and ACF2:

- AMD2RACF for RACF
- AMD2TS for Top Secret
- AMD2ACF2 for ACF2

In addition, you can specify either the name of a user-written exit or NONE (the default).

For information about building an exit program, see “Preparing to code your routine.”

Host request library ID

The *Host request library ID* specifies the qualifier used for the AMD2-HOSTLIB table. The qualifier name can contain up to 25 characters. The default value is SYBASE.

Temp storage queue type

This option indicates the temporary storage queues used for results sent to the client. Specify one of the settings shown in the following table.

Table 4-1: Types of temporary storage queues

Setting	**Type of storage**	**DB2 UDB Option for CICS action**
MAIN	Main	Writes all results to main temporary storage before sending results to the client. After results are sent, temporary storage queues are deleted unless the mainframe debug option is set on the LAN using the LAN set command.
AUX	Auxiliary	Writes all results to auxiliary temporary storage before sending results to the client. After results are sent, temporary storage queues are deleted unless the mainframe debug option is set on the LAN using the LAN set command.
NONE **Note** Temporary storage is only used for data going to a client.	None (default)	Sends results to the client without writing them to temporary storage. This decreases the wait for results and the overall transmission time, especially for large transfer operations. This setting also saves temporary storage space, which would be needed for results. This setting has the following disadvantages: • If a processing error occurs, data has already been sent to the client. • This setting cannot be used with a result exit. • When accessing DB2, DB2 threads and locals can be held longer because they are allocated for the time required to transmit the results over the APPC link, rather than the time required to write the results to temporary storage.

Estimating temporary storage

The amount of temporary storage required is determined by the size of the result set to be returned. For example, if 150,000 rows of 1200 bytes are to be returned, 180,000,000 bytes of temporary storage are required to hold the result set.

- If *Temp storage queue type* is set to MAIN, this storage must be available in main memory.
- If *Temp storage queue type* is set to AUX, this storage must be available on DASD.

Determining main memory storage

CICS keeps an 8-byte temporary storage ID in main memory for each temporary storage record written to either main memory or DASD (auxiliary storage). Use the following equation to determine the number of bytes of main memory storage you need:

N x *ID* = *M*

where:

- *N* is the number of records written to temporary storage.
- *ID* is the 8-byte storage ID.
- *M* is the number of bytes of main storage required.

DateTime results type

The *DateTime results type* specifies the format that the DB2 UDB Option for CICS uses to return DB2 timestamp, time, and date result columns to TRS or non-DB2 access service connections. Three values are valid:

- DB2 (the default) returns timestamps in the standard DB2 timestamp format of *YYYY-MM-DD-HH.MM.SS.ssssss*, where *YYYY* indicates the year, *MM* indicates the month, *DD* indicates the day, *HH* indicates the hour, *MM* indicates the minute, and *SS.ssssss* indicates the seconds. For example, a DB2 value might return 2002-05-14-1.42.22.123456.
- TRO returns timestamps in a format compatible with ASE/CIS communication with DirectConnect TRS using the ASE/CIS Access Server Class.

TRO format is *YYYYMMDD HH:MM:SS:ss*, where the abbreviations are similar to the DB2 abbreviations. Using the example from the DB2 value, the TRO value returns 20020514 1:42:22:12.

- ISO returns timestamps in the ISO (International Standards Organization) format of *YYYY-MM-DD hh.mm.ss*, where the abbreviations are similar to the DB2 abbreviations. Following the above examples, an ISO value might return 2002-05-14 13.42.22.

Send UDT

Send UDT (User Data Type) determines whether the UDT is returned for each column description. The value of *Send UDT* can be either YES or NO. The default value is YES.

Long Varchar

Long Varchar specifies the data type of results returned to the client application for Long Varchar, either as Long Varchar or Text. The value of *Long Varchar* is either LVC or TEXT. The default value is LVC.

RPC option

RPC option specifies how the DB2 UDB Option for CICS should respond when invoked as an RPC. Two values are valid:

- LANG specifies that the provided RPC parameters are concatenated to form a language event.
- DB2SP specifies that the RPC call is formatted into a DB2 stored procedure call.

DB2 caching option

DB2 caching option enables the DB2 UDB Option to use DB2 caching by converting all constants in a SQL language statement to host variables. DB2 uses a cached DBRM for any SQL statements that exactly match a cached statement. This results in more CPU usage in the DB2 UDB Option but less in DB2. The DB2 caching option parameter has the following settings:

- YES converts all constants in SQL language statements to host variables.
- NO leaves SQL language statements as they are.

Tracing option

Tracing option allows internal program tracing to assist Sybase Technical Support in researching problems. Turning on the tracing option results in substantially higher CPU and time overhead.

- YES activates all internal program tracing for this transaction.
- NO disables tracing for this transaction.

Results are written to a temporary storage queue, the name of which consists of “CE” and the first 6 bytes of the user ID.

Using the DB2 UDB Option for CICS with utf8

The DB2 UDB Option for CICS can be configured to use utf8 for communication with DB2 and clients. This configuration gives the greatest flexibility for using different character sets when communicating with DB2 since most character sets can be converted to and from utf8. All the accessed DB2 tables should be single-byte EBCDIC tables or should have Unicode encoding schema.

To configure the DB2 UDB Option for CICS to use utf8, set the following configuration parameters in the Server Option for CICS SYGWMCST macro of the SYGWXCPH options module:

```
USEIBMUNICODE=Y
CHARSETSRV=utf8
```

You must also bind the AMD2DB2 and AMD2DBA DBRM packages with “encoding(unicode)” since all communication between the DB2 UDB Option for CICS and DB2 are now in utf8.

Note To retain MDI legacy RSP feature functionality for the Server Option for CICS, do not set the CHARSETSRV parameter to “utf8” if RSPs are run through the DirectConnect for z/OS Option or if they are run as “use procedure” in a gateway-less environment. The CHARSETSRV parameter must be set to iso_1 for RSPs to work. RSPs run as RPCs through a DirectConnect for z/OS Option TRS or in a gateway-less environment are not affected.

CHAPTER 5

Customizing the DB2 UDB Option for CICS Messages

The AMD2 message table in the DB2 UDB Option for CICS determines the messages returned to the client application. You can modify the AMD2 message table to customize messages using single-byte character sets (SBCSs) for your site, or to accommodate new SQLCODEs generated by updated versions of DB2.

Note You cannot customize Japanese or other double-byte character set (DBCS) languages, which are handled by the Japanese Conversion Module. For more information, see the installation chapter for the Japanese Conversion Module in the Mainframe Connect Server Option for CICS *Installation and Administration Guide*.

This chapter describes the AMD2 message table and explains how to make changes and additions to the existing messages. It covers the following topics:

Understanding the AMD2 message table

The AMD2 message table consists of two separate tables that share a common property: the message number. These two separate tables, the *translate table* and the *text table*, function together to return messages to the client.

- The *translate table* associates DB2 SQLCODEs with message numbers and sets the attributes of errors.

You can edit the translate table to add messages or to change their attributes. (Attributes include the message number, the error severity, and so on.) As shipped, the AMD2 message table contains entries for SQLCODEs returned by Version 3.1 of DB2.

- The *text table* associates message numbers with message text.
  You can edit the text table to change the text of messages. You can also expand it to provide message text in additional languages. Message text for U.S. English, German and French is provided with the product.

The AMD2 message table is generated during installation as one module, AMD2CAMX, which contains all the AMD2 messages and message attributes. AMD2CAMX is built with two macros, SYRTMAMX and SYRTMAMT, and it is composed of two CSECTS, SYRTMAMX and SYRTMAMT. The following table shows the members of the translate and text tables.

Table 5-1: Message table members

Message table	LOAD module	MACRO library	SOURCE library
Translate table	AMD2CAMX	SYRTMAMX	AMD2CAMX
Text table	AMD2CAMX	SYRTMAMT	AMD2CAMX

Translate table

The translate table is the first table used in processing AMD2 messages. When DB2 returns a SQLCODE, AMD2 searches the translate table by the SQLCODE to determine the resulting output message number (MSGNR) and other attributes associated with the SQLCODE. For non-DB2 errors, AMD2 searches the translate table by MSGNR to determine the other message attributes associated with the error.

Example of translate table source code

The editable SOURCE library member (AMD2CAMX) of the translate table contains a series of SYRTMAMX macro entries, such as those shown in the following example:

```
SYRTMAMX TYPE=ENTRY,              *
     SQLCODE=+802,                *
     MSGNR=30001,                 *
     LEVEL=11,                    *
     STATEC=22,                   *
     STATESC=3,                   *
     LOG=N,                       *
```

```
          SQLFATAL=N                   *

     SYRTMAMX TYPE=ENTRY,             *
          SQLCODE=-007,               *
          MSGNR=102,                  *
          LEVEL=11,                   *
          STATEC=31,                  *
          STATESC=503,                *
          LOG=N,                      *
          SQLFATAL=N
```

Translate table properties

Each SYRTMAMX macro entry in the translate table consists of the following DB2 error message properties:

TYPE

The TYPE property indicates whether the SYRTMAMX macro entry is the initial (INITIAL) or final (FINAL) entry in the translate table, or is one of the entries in between (ENTRY). Therefore, TYPE=INITIAL occurs once at the beginning of the translate table; TYPE=FINAL occurs once at the end of the table. Examples of the TYPE=INITIAL and TYPE=FINAL are shown as follows:

```
     SYRTMAMX TYPE=INITIAL
   ... (additional entries not shown)
          SYRTMAMX TYPE=FINAL
```

Warning! Do not change the INITIAL or FINAL value.

SQLCODE

The SQLCODE property contains the SQLCODE, a numeric status indicator returned by DB2. For messages that DB2 does not generate, such as internal error messages that AMD2 generates, the value of SQLCODE is always 0. This property contains 3 digits and a plus (+) or minus (-) sign.

MSGNR

The text table uses the message number to identify the appropriate message to send to the client. It represents a single SQLCODE or a group of SQLCODEs about similar situations.

Warning! Do not change the number of non-DB2 messages (those with SQLCODE equal to 0 and MSGNR greater than or equal to 32000).

LEVEL

The LEVEL property indicates the severity of the error. It corresponds to the following Adaptive Server severity levels:

01–10	Status information
11–16	Client application errors
17–18	Non-fatal software or hardware errors
19–24	Fatal system errors

STATEC

The STATEC class property, a 2-byte field, provides more information about DB2 errors. For non-DB2 errors, the value is 0.

STATESC

The STATESC subclass property, a 3-byte field, provides more information about DB2 errors. For non-DB2 errors, the value is 0.

LOG

The LOG (logging instruction) property indicates whether to log the error in the *AMD2LOG* file. Sybase recommends logging serious errors. To turn logging on, specify the Y value. To turn logging off, specify the N value.

SQLFATAL

The SQLFATAL (fatal error) property indicates whether AMD2 accepts more client requests in the current invocation.

- Y means that yes, the error is fatal, and therefore, no further requests are accepted from the current client connection.
- N means that no, the error is not fatal, and requests are processed as usual.

Text table

The text table is the part of the AMD2 message table that determines the text of a message by connecting message text with message numbers. For each language, there is a message number and associated text that corresponds to each message number in the translate table. After translate table processing, AMD2 searches the text table by language and message number to determine the message text to send to the client.

Example of text table source code

The editable SOURCE library member AMD2CAMX in the text table contains a series of SYRTMAMT entries, one for each language. Each SYRTMAMT entry contains a listing of message numbers and associated text.

Portions of the text table are shown in the following example:

```
* Message text                                     *
*--------------------------------------------------*
  SYRTMAMT TYPE=INITIAL
*
  SYRTMAMT TYPE=ENTRY,LANG=us_english,             *
```

```
   102,'Incorrect syntax (DB2 %s failed: %S)',    *
   103,'String is too long (DB2 %s failed: %S)',  *
   105,'Unclosed quote (DB2 %s failed: %S)',      *
        ... (additional entries not shown)
*
  SYRTMAMT TYPE=ENTRY,LANG=german,                *
   102,'Falsche Syntax (DB2 %s                    *
        fehlgeschlagen: %S)',                     *
   103,'Zeichenkette zu lang (DB2 %s              *
        fehlgeschlagen:%S)',                      *
   105,'Schlieendes Anfhrungszeichen fehlt        *
        (DB2 %s * fehlgeschlagen:%S)',            *
            ... (additional entries not shown)
*
   SYRTMAMT TYPE=ENTRY,LANG=french,                *
    102,'Syntaxe incorrecte (chec de DB2 %s : %S)',*
    103,'Chane trop longue (chec de DB2 %s : %S)', *
    105,'Guillemet fermant manquant(chec de DB2    *
         %s :%S)',                                 *
     ... (additional entries not shown)
*
   SYRTMAMT TYPE=FINAL
```

National languages

As shipped, the text table contains SYRTMAMT entries to supply message text in three languages: U.S. English, German, and French. Each SYRTMAMT entry contains a complete set of messages in the language specified by the LANG property located in the first line of the entry.

If the text table does not contain a LANG property to match the language identification passed to AMD2, the default is the U. S. English message set. For information about how languages are defined, see the customization options appendix in the Mainframe Connect Server Option for CICS *Installation and Administration Guide* and the DirectConnect documentation.

Note A default message (“Error 33231” on page 169) returns whenever a problem with the AMD2 message table occurs. This message is always sent in U.S. English, regardless of the text coded in the text table.

Message variable indicators

The message the client application receives contains the words specified in the text table and, in some cases, the contents of one or more variable indicators: *%s*, *%d*, and *%S*. The words and the variables are specified in the text table.

AMD2 supplies the content of the message variable indicators before a message is returned to the client. Regardless of the language of the message text, the content of the variables is always in U.S. English.

AMD2 replaces the variable indicators with the variable values shown in the following table.

Table 5-2: Message variable indicators and values

Message variable indicator	Replaced by message variable value
%s	String message variable value indicating the DB2 function last executed, usually PREPARE.
%d	Numeric message variable value used for any integer values, such as an error number or the position of an error.
%S	Messages returned by DB2 routine DSNTIAR, including SQLCA message tokens and SQLCODEs.

Changing the AMD2 message table

You can customize and update AMD2 messages by changing the AMD2 message table. For example, you can:

- Add entries for new SQLCODEs
- Add or change messages for a parse exit routine
- Customize message text
- Eliminate messages
- Change message logging

Add entries for new SQLCODEs

DB2 versions later than 4.1 return SQLCODEs without corresponding entries in the translate table. In this circumstance, AMD2 assigns a generic error message (“Error 33231” on page 169) and sends the associated text from the text table for that message number. To provide clients with more than the generic message, add new entries to the AMD2 message table for each new DB2 SQLCODE.

There are two options for translating new SQLCODEs to client messages. New SQLCODEs can be:

- Mapped to existing DB2 UDB Option for CICS messages (“Error 30000” on page 156, “Error 30001” on page 157, and “Error 30002” on page 157).
- Mapped to messages that correspond to Adaptive Server messages. These Adaptive Server messages may exist in the AMD2 message table, or you may need to create new entries for these messages.

The changes to the AMD2 message table for the preceding two options are shown in the following table.

Table 5-3: Adding entries for new SQLCODEs

Option	Message number (MSGNO=)	Translate table changes	Text table changes	For coding the changes, see
Map to existing DB2 UDB Option messages.	Assign number by severity level: 30000–Error 30001–Warning 30002–Confirmation	Add an SYRTMAMX entry for each new message.	No change.	“Adding a translate table entry” on page 44.
Map to Adaptive Server messages.	Use corresponding Adaptive Server message number.	Add an SYRTMAMX entry for each new message.	• If message is already in text table, no change is needed. • If message is not in text table, add the message number and the text of Adaptive Server message to SYRTMAMT entry for each language in use.	“Adding a translate table entry” on page 44. “Adding a message to the text table” on page 46.

Add or change messages for a parse exit routine

If you customize the DB2 UDB Option for CICS by adding a parse exit routine, you can add additional messages to the AMD2 message table to report on the processing of your routine. A sample parse exit routine ships with the DB2 UDB Option for CICS with a corresponding message number 33400 (“Error 33400” on page 179) in the AMD2 message table.

You can modify the sample routine entries in the translate and text tables or you can create new entries. The changes to the AMD2 message table for these two options are shown in the following table.

Table 5-4: Adding or changing messages for a parse exit routine

Option	Message number (MSGNO=)	Translate table changes	Text table changes	For coding the changes, see
Modify the sample parse exit routine message.	33400.	Change SYRTMAMX properties as appropriate.	Replace message text as appropriate in the SYRTMAMT entry for each language in use.	“Modifying a translate table entry” on page 45. “Modifying a message in the text table” on page 46.
Create new messages.	See “Selecting an error message” on page 106 for available numbers.	Add an SYRTMAMX entry for each new message.	Add messages to the SYRTMAMT entry for each language in use.	“Adding a translate table entry” on page 44. “Adding a message to the text table” on page 46.

Customize message text

If the existing message text does not meet your needs, you can change the text of specific messages or translate all the messages into a different language. However, the following messages or parts of messages cannot be customized or translated:

- “Error 33231” on page 169 always returns with as-shipped U.S. English text, regardless of what is coded in the text table.
- The message variable indicator strings, *%s* and *%S*, in the text table are always replaced with values that contain U. S. English text. Many of these values are DB2 commands or messages, which are always returned in U.S. English.

The following table shows the changes to the AMD2 message table.

Table 5-5: Customizing message text

Option	Message number (MSGNO=)	Translate table changes	Text table changes	For coding the changes, see
Modify existing message text.	The message number is not changed.	No change.	Replace existing text with new text in the SYRTMAMT entry for each language in use.	"Modifying a message in the text table" on page 46.
Translate messages.	No new numbers are assigned, but text for all numbers must be translated.	No change.	Add an SYRTMAMT entry with translated message text.	"Adding a language to the text table" on page 46.

Eliminate messages

The DB2 UDB Option for CICS sends confirmation, warning, and error messages from DB2 to the client application. You can change the AMD2 message table to intercept those messages before they are sent, thus reducing message traffic by eliminating some of the messages. To prevent the sending of specific messages, you can make changes to the AMD2 message table as shown in the following table.

Table 5-6: Eliminating messages

Option	Message number (MSGNO=)	Translate table changes	Text table changes	For coding the changes, see
Prevent a message from being sent to a client application.	Change the message number to 0 (MSGNO=0).	Change the value of the MSGNO property to 0 (MSGNO=0).	No change.	"Modifying a translate table entry" on page 45.

Change message logging

As shipped, the DB2 UDB Option for CICS writes messages to the *AMD2LOG* file based on the preassigned values of the translate table's LOG property. You can control which messages are written to the error log file *AMD2LOG* by changing the value of the LOG property in the translate table for each message involved. To change message logging, make changes to the AMD2 message table as shown in the following table.

Table 5-7: Changing message logging

Option	Message number (MSGNO=)	Translate table changes	Text table changes	For coding the changes, see
Write a message to the error log file.	No change.	Change the value of the LOG property to Y.	No change.	"Modifying a translate table entry" on page 45.
Prevent a message from being written to the error log file.	No change.	Change the value of the LOG property to N.	No change.	"Modifying a translate table entry" on page 45

Coding AMD2 message table changes

To code changes to the AMD2 message table, determine whether you must change the translate table, the text table, or both, then proceed as shown in the following table.

Table 5-8: Changing the AMD2 message table

To modify	Follow the instructions in
Translate table	"Change the translate table" on page 44
Text table	"Change the text table" on page 45

Note After making the changes, assemble and link the message table.

Change the translate table

Changes to the translate table are made in the SOURCE library member AMD2CAMX. You can modify existing SYRTMAMX entries or you can add new ones; do not delete entries.

Adding a translate table entry

To add a SYRTMAMX macro entry to the translate table, follow these steps:

1 Copy an existing SYRTMAMX macro entry.

2 Add the copied entry to the translate table.

3 Modify the property values as needed.

For help in determining property values, see the explanation in "Translate table properties" on page 37.

For additional information about assigning message numbers, see "Add entries for new SQLCODEs" on page 41 and "Add or change messages for a parse exit routine" on page 42.

Modifying a translate table entry

To modify an existing SYRTMAMX entry, follow these steps:

1 Locate the target entry.

2 Change the property values as necessary.

For example, if you do not want messages returned to the client application for certain DB2 SQLCODEs, set the message number property to 0 (MSGNO=0).

Note Deleting a SYRTMAMX entry generates default messages or errors; it does not reduce the number of messages sent to the client application. To prevent the sending of certain messages, modify the property values (see "Eliminate messages" on page 43) instead of removing the entry.

Change the text table

Changes to the text table are made in the SOURCE library member AMD2CAMX. You can modify existing SYRTMAMT macro entries by adding messages or changing message text, but you should not delete messages, because then messages will not be returned for the error codes. You can also add SYRTMAMT entries to provide message text in additional languages.

Rules for text table entries

The following rules apply to text table messages:

- Message number — Do not change the message numbers of existing messages because those numbers correspond to MSGNO values in translate table entries. For new text table entries, be sure the message number added to the text table has a corresponding message added to the translate table.

- Message text string — You can replace existing text or add new text. The text string, including message variable indicators, can be up to 255 characters in length. After AMD2 replaces the variable indicators with variable values, the message text returned to the client application never exceeds 512 characters.

Modifying a message in the text table

To modify the text of existing text table messages, type over the target message text with your new text, following the guidelines in “Rules for text table entries” on page 45.

Adding a message to the text table

To add a message to the text table, follow the format of other messages in the editable member AMD2CAMT, and observe the guidelines in “Rules for text table entries” on page 45.

To avoid message processing errors, be sure to add any new messages to all the SYRTMAMT entries in use at your installation. For example, if both United States and German customers can use the DB2 UDB Option for CICS, add all new messages to both the U.S. English and the German message sets. If a message number exists in a SYRTMAMX entry but not in the SYRTMAMT entry for the specified language, “Error 33231” on page 169, in U.S. English, is sent to the client application.

Adding a language to the text table

To translate message text into a different language, add an SYRTMAMT entry, containing an entire set of messages, to the SOURCE library member AMD2CAMX. To add the entry, copy and modify an existing SYRTMAMT entry (U.S. English, French, or German) as follows:

1. Place the copied SYRTMAMT entry in the text table just above the following line:

   ```
   SYRTMAMT TYPE=FINAL
   ```

2. Change the LANG property to match the language identification passed to AMD2 for the target language.

   The first line of the new entry should look like this:

   ```
   SYRTMAMT
   TYPE=ENTRY,LANG={us_english|german|french},
   ```

3 Replace the message text with a translation in the target language.

- For information about the variable indicators, *%s*, *%d* and *%S*, see “Message variable indicators” on page 40.
- Be sure the translated text follows the guidelines in “Rules for text table entries” on page 45.

CHAPTER 6 Using Stored Procedures

This chapter describes the role and use of stored procedures with the DB2 UDB Option for CICS and covers the following topics:

Understanding stored procedures

A stored procedure is an application stored in an executable state and used by a server to access mainframe data. Stored procedures used with the DB2 UDB Option for CICS can be classified in three categories: remote procedure calls (RPCs), remote stored procedures (RSPs), and DB2 stored procedures.

- *Remote procedure call* - An RPC is a stored procedure accessed indirectly through a client request and executed on a Server Option server. An RPC is a user-written program using the Server Option API.
    - *Catalog stored procedure* - A CSP is a set of RPCs that return system and table information and provide an interface for accessing system tables in a variety of database management systems.
- *Remote stored procedure* - An RSP, like an RPC, is accessed indirectly through a client request. However, RSPs use the old MDI API, not the Server Option API.
- *DB2 stored procedure* - A DB2 stored procedure is a user-written application invoked by the DB2 CALL command and executed inside DB2 address space.

The remainder of this chapter consists of the following subsections:

- DB2 stored procedures
- Remote stored procedures
- Remote procedure calls

DB2 stored procedures

The DB2 UDB Option for CICS supports the use of DB2 stored procedures in both a two-tier, gateway-less, and a three-tier, gateway-enabled environment.

❖ **Executing a DB2 stored procedure:**

1. Define an RPC that has the same case-sensitive name as the DB2 stored procedure you want to use.
2. Make sure the CICS transaction name to which the RPC is mapped has a DB2 UDB Option setting for the RPC option of the DB2 stored procedure.
3. Execute the defined RPC with parameters matching the definition of the DB2 stored procedure.

For information on writing and using DB2 stored procedures, refer to the appropriate IBM documentation.

Remote stored procedures

Remote stored procedures (RSPs) are similar to RPCs, but they use the old MDI API, not the Server Option API. RSPs are normally invoked as language statements through the DirectConnect DB2 access service.

To invoke an RSP as an RPC in a two-tier, gateway-less environment or through DirectConnect Transaction Router Service, the user must alter the RSP code.

Running RSPs as RPCs

To run an RSP as an RPC, you must modify the RSP to include the RPSETUP and RPDONE commands.

For more information on developing and using RSPs, see the Mainframe Connect Server Option *Programmer's Reference for Remote Stored Procedures*.

Remote procedure calls

Remote procedure calls (RPCs) are stored procedures executed indirectly through the request of a client application. Sets of RPCs used to return system information, table information, and capabilities data can be grouped into a catalog stored procedure (CSP). A CSP serves as a uniform catalog interface for accessing the system tables of different database management systems. Sybase supplies CSPs with the DB2 UDB Option for CICS. These CSPs can be used with JDBC or ODBC.

Catalog stored procedures

To obtain information about database objects, you need to access the database catalog. Catalog stored procedures (CSPs) provide this catalog access. This section describes how to use CSPs to access the DB2 UDB catalog and covers the following topics:

- Using CSPs
- Supported CSPs
- Using CSPs to access non-catalog DB2 information
- CSPs in a non-DB2 environment

Using CSPs

This section describes the use of CSPs, coding instructions that apply to CSPs, and the use of wildcard-character search patterns.

Using CSPs

The catalog structures for DB2 UDB and Sybase Adaptive Server are different. If you have client applications written to access the Adaptive Server Enterprise catalog, you may need to re-code the client application queries to send those queries directly to the DB2 UDB system tables. To avoid modifying your database-specific applications, you can use CSPs to access catalog information. The CSPs supplied with the DB2 UDB Option for CICS are compatible with the catalog interface for ODBC or JDBC.

Coding instructions

This section includes general coding information that applies to all CSPs.

Parameters

CSPs have optional and required parameters. Required parameters must have values supplied; optional parameters default to predefined values.

The following rules apply to CSP parameters:

- Both positional and named parameters are supported, but not in the same statement.
- Parameter values can be enclosed in double quotes (“”). Parameter values enclosed in quotes must be in the correct case for the target.
- Object names (table names, column names, and index names) can be created using lowercase letters. The target database automatically converts object names to uppercase unless the object names are enclosed in double quotes.

Syntax

Using the DirectConnect DB2 access service, a client application can initiate a CSP by issuing the following statement:

```
exec sp_name parm1, parm2, . . .
```

where:

- *sp_name* is the name of the stored procedure (for example, sp_columns).
- *parm1* and *parm2* are parameter values required or desired for that stored procedure.

Using the DirectConnect Transaction Router Service or from a two-tier, gateway-less environment, a client application can initiate a CSP by issuing the following statement:

```
exec|execute sp_name parm1, parm2, . . .
```

where:

- *sp_name* is the name of the stored procedure (for example, sp_columns).
- *parm1* and *parm2* are parameter values required or desired for that stored procedure.

Coding examples

You can execute CSPs with a language command or through an RPC event.

You can specify the parameters for a CSP in one of the following forms:

- Supply all of the parameters:

  ```
  exec sp_columns publishers, “dbo”, “pubs2”, “pub_id”
  ```

- Use “null” or a comma as a placeholder:

  ```
  exec sp_columns publishers, null, null,”pub_id”
  exec sp_columns publishers, , , “pub_id”
  ```

- Supply one or more parameters in the following form:

  ```
  @parameter_name = value
  ```

  For example, to find information about a particular column, issue the following statement:

  ```
  exec sp_columns @table_name = publishers,
  @column_name = "pub_id"
  ```

  The parameter names in the syntax statement must match the parameter names defined by the CSP.

  You cannot use this named parameter form if you process a CSP as an RPC event.

Table name, owner, and qualifier parameters

This section explains how the parameters *table_name*, *table_owner,* and *table_qualifier* are used in this product:

- *table_name* is the name of the database object about which you want to retrieve catalog information.

- *table_owner* is the owner of the database object about which you want to retrieve catalog information.

Wildcard-character search patterns

The percent (%) wildcard character can be used in parameters that allow wildcard-character search patterns. This wildcard represents any string of zero or more characters.

The following table shows some examples of the percent (%) wildcard character and its use:

Table 6-1: Wildcard character examples

Sample string	Matches
%A%	All names that contain the letter "A"; for example, A, AT, CAT
%	All names

Escape character

To use a wildcard character as a literal, precede it with an @ (at) sign. If the parameter normally accepts the wildcard character, you can mix the percent (%) wildcard character with escaped wildcard characters (@%) interpreted as literals. If the parameter does not accept the wildcard character, an @ (at) sign must precede the wildcard character to use the character as a literal.

Supported CSPs

The following table shows the supported CSPs and the information that each CSP retrieves.

Table 6-2: Supported CSPs

CSP	Information retrieved by the CSP
sp_capabilities	SQL capabilities
sp_column_privileges	Column privilege information for one table
sp_columns	Column descriptions for a table
sp_databases	List of available databases
sp_datatype_info	Datatype descriptions
sp_fkeys	Foreign and primary key relationships
sp_mda	Server metadata
sp_pkeys	Primary key information for a single table
sp_server_info	Server terms, limits, and capabilities
sp_special_columns	Additional column information
sp_sproc_columns	Attributes of procedures input and return parameters
sp_statistics	Statistics and indexes for one table
sp_stored_procedures	List of available procedures
sp_table_privileges	Table privilege information for one table
sp_tables	List of aliases, synonyms, tables, views, and system tables
sp_thread_props	Thread properties

The following sections provide descriptions, syntax, parameters, and usage for the supported CSPs.

sp_capabilities

Description

Returns the SQL capabilities of the server.

Syntax

```
sp_capabilities
```

Parameters

This procedure does not use or allow parameters.

Usage

- The result set contains information that allows applications to successfully interact with the server during normal query processing.

Results

The following table shows the result set:

Table 6-3: Result set for sp_capabilities

Column name	Datatype	Description
ID	int	Capability ID
CAPABILITY_NAME	char(30)	Capability name
VALUE	int	Capability value
DESCRIPTION	char(128)	Capability description

The following table shows values for the capabilities.

Table 6-4: sp_capabilities information

ID	Capability	Value description
101	SQL syntax	1=Sybase Transact-SQL® supported 2=DB2 SQL supported
102	Join handling	0=Unsupported 1=No outer join supported 2=Transact-SQL support 3=Oracle supported
103	Aggregate handling	0=Unsupported 1=ANSI supported 2=All functions
104	AND predicates	0=Unsupported 1=Supported
105	OR predicates	0=Unsupported 1=Supported
106	LIKE predicates	0=Unsupported 1=ANSI-style supported 2=Transact-SQL supported
107	Bulk insert handling	0=Unsupported 1=Supported
108	Text and image handling	0=Unsupported 1=Text, no textptr 2=Text and textptr
109	Transaction handling	0=Unsupported 1=Local supported 2=Two-phase commit supported
110	Text pattern handling	0=Unsupported 1=Pattern (text) supported
111	order by	0=Unsupported 1=Supported
112	group by	0=Unsupported 1=ANSI supported 2=Transact-SQL supported
113	Net password encryption	0=Unsupported 1=Supported
114	Object case sensitivity	0=Case insensitive 1=Case sensitive
115	distinct	0=Unsupported 1=Supported
116	Wildcard escape	0=Unsupported Non-zero=Escape_char(s)

ID	Capability	Value description
117	Union handling	=Unsupported 1=Supported
118	String functions	0=Unsupported 1=Substring supported 2=Oracle subset supported 3=Transact-SQL supported
119	Expression handling	0=Unsupported 1=ANSI supported 2=Transact-SQL supported
120	Character truncation	0=Fixed length character parameters may contain trailing blanks 1=Fixed length character parameters will not contain trailing blanks
121	Language events	0=Unsupported 1=Transact-SQL DML without datetime in the where clause supported 2=Transact-SQL DML supported
122	Date functions	0=Unsupported 1=Transact-SQL date functions supported
123	Math functions	0=Unsupported 1=Oracle functions supported 2=Transact-SQL math functions supported
124	Transact-SQL convert functions	0=Unsupported 1=Supported
125	Transact-SQL delete/update	0=Sybase extensions not supported 1=Sybase extensions supported
126	Insert/select handling	0=Unsupported 1=Supported
127	Subquery handling	0=Unsupported 1=Supported
128	IN/NOT IN support	0=Unsupported 1=Supported
129	CASE support	0=Unsupported 1=Supported

sp_column_privileges

Description

Returns column privilege information for a single database object.

Syntax

sp_column_privileges *table_name* [, *table_owner*]
[, *table_qualifier*] [, *column_name*]

Parameters

table_name is the name of the table. Wildcard-character search patterns and aliases are not supported. Views are supported but do not include alter or index privileges.

table_owner is the name of the table owner. Wildcard-character search patterns are not supported.

table_qualifier is ignored. Leave blank or set to NULL.

column_name is the name of the column for which you want privilege information. Use wildcard-character search patterns to request information about more than one column. Leave blank or set to NULL to request information about all columns in the table or tables.

Usage

- This function corresponds to the ODBC function SQLColumnPrivileges.
- Information is based on the SYSCOLAUTH, SYSCOLUMNS, and SYSTABAUTH system catalog tables.

Results

sp_column_privileges returns one row for each privilege a user has on a column in a table. Results are ordered by the following columns:

- TABLE_OWNER
- TABLE_NAME
- COLUMN_NAME
- PRIVILEGE

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_column_privileges.

Table 6-5: Result set for sp_column_privileges

Column name	Datatype	Description
TABLE_QUALIFIER	varchar (128)	Always NULL.
TABLE_OWNER	varchar (128)	Authorization ID.
TABLE_NAME	varchar (128) NOT NULL	Name of the object about which privilege information is returned.
COLUMN_NAME	varchar (128) NOT NULL	Column name.
GRANTOR	varchar (128)	Identifies the user who granted this privilege.
GRANTEE	varchar (128) NOT NULL	Identifies the user to whom this privilege was granted.
PRIVILEGE	varchar (128) NOT NULL	Identifies the privilege granted to the grantee on this column as one of the following values: • SELECT if the grantee is authorized to select rows in the associated object. • UPDATE if the grantee is authorized to insert and update rows in the associated object.
IS_GRANTABLE	varchar (3)	Indicates whether the grantee is authorized to grant privilege on this column to other users; always NULL.

sp_columns

Description

Returns information about the type of data that can be stored in one or more columns.

Syntax

sp_columns *table_name* [, *table_owner*]
 [, *table_qualifier*] [, *column_name*]

Parameters

table_name is the table name. Use the wildcard character to request information about more than one table. Aliases are not supported.

table_owner is the owner of the database object about which column information is requested. Use the wildcard character to request information about tables owned by more than one user. If you do not specify a table owner, sp_columns looks first for tables owned by the current user and then for tables owned by the database owner.

table_qualifier is ignored. Leave blank or set to NULL.

column_name is the name of the column for which you want information. Use the wildcard character to request information about more than one column. Leave empty or set to NULL to request information about all columns in the table or tables.

Usage

- If *column_name* is provided, sp_columns returns information only for the column or columns that match.
- This function corresponds to the ODBC function SQLColumns.
- Information is based on the SYSCOLUMNS and SYSSYNONYMS system catalog tables.

Results

sp_columns returns one row containing a description of each column in a table. Results are ordered by the following columns:

- TABLE_OWNER
- TABLE_NAME

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_columns.

Table 6-6: Result set for sp_columns

Column	Datatype	Description
TABLE_QUALIFIER	varchar(128)	Always NULL
TABLE_OWNER	varchar(128)	Table owner identifier
TABLE_NAME	varchar(128) NOT NULL	Table name
COLUMN_NAME	varchar(128) NOT NULL	Column name
DATA_TYPE	smallint NOT NULL	Integer code for the ODBC datatype
TYPE_NAME	varchar(128) NOT NULL	String representing the datatype name in the target database
PRECISION	int	Number of significant digits of the column on the target database
LENGTH	int	Length of the column in bytes
SCALE	smallint	Number of digits to the right of the decimal point

Column	Datatype	Description
RADIX	smallint	Base for numeric types
NULLABLE	smallint NOT NULL	Indicates whether the column accepts NULL values: 0 SQL_NO_NULLS if the column does not accept NULL values 1 SQL_NULLABLE if the column accepts NULL values 2 SQL_NULLABLE_UNKNOWN if it is not known if the column accepts NULL values
REMARKS	varchar(254)	A description of the column
SS_DATA_TYPE	smallint	The Adaptive Server Enterprise datatype name
COLID	smallint	The column ID number
REMOTE_DATA_TYPE	int	An integer representing the underlying target database datatype (composite value)

ODBC datatypes

The following table describes the DB2 UDB datatypes and matching ODBC integer identifiers that are returned in the TYPE_NAME and DATA_TYPE columns of the sp_columns, sp_datatype_info, sp_special_columns, and sp_sproc_columns result sets.

Table 6-7: ODBC datatypes

DB2 UDB datatype	Target datatype maximum physical length	ODBC type	ODBC integer ID	DB2 UDB datatype description
CHARACTER() FOR BIT DATA	254	SQL_BINARY	-2	Fixed length character for bit data
VARCHAR() FOR BIT DATA	254	SQL_VARBINARY	-3	Variable length character for bit data
LONG VARCHAR FOR BIT DATA	32714	SQL_LONGVARBINARY	-4	Variable length character for bit data
CHARACTER()	254	SQL_CHAR	1	Fixed length character
VARCHAR()	254	SQL_VARCHAR	12	Variable length character
LONG VARCHAR()	32714	SQL_LONGVARCHAR	-1	Variable length character
CHARACTER() FOR MIXED DATA	254	SQL_BINARY	-2	Fixed length character (DBCS or SBCS)

DB2 UDB datatype	Target datatype maximum physical length	ODBC type	ODBC integer ID	DB2 UDB datatype description
VARCHAR() FOR MIXED DATA	254	SQL_VARBINARY	-3	Variable length character (DBCS or SBCS)
LONG VARCHAR() FOR MIXED DATA	32714	SQL_LONGVARBINARY	-4	Variable length character (DBCS or SBCS)
GRAPHIC()	127	SQL_BINARY	-2	Fixed length graphic (DBCS)
VARGRAPHIC()	127	SQL_VARBINARY	-3	Variable length graphic (DBCS)
LONG VARGRAPHIC	16357	SQL_LONGVARBINARY	-4	Variable length graphic (DBCS)
SMALLINT	2	SQL_SMALLINT	5	2-byte binary integer
INTEGER	4	SQL_INTEGER	4	4-byte binary integer
REAL	4	SQL_REAL	7	4-byte floating point
FLOAT()	4	SQL_REAL	7	4-byte floating point with a precision less than 22
FLOAT()	8	SQL_DOUBLE	8	8-byte floating point with a precision equal to or greater than 22
DOUBLE PRECISION	8	SQL_DOUBLE	8	8-byte floating point
DECIMAL()	31	SQL_DECIMAL	3	Packed decimal number
NUMERIC	31	SQL_NUMERIC	2	Zoned decimal number
DATE	10	SQL_DATE	9	Date
TIME	8	SQL_TIME	10	Time
TIMESTAMP	26	SQL_DATETIME	11	Timestamp

REMOTE_DATATYPE

The REMOTE_DATATYPE column contains a 32-bit composite datatype value that represents the target database datatype. The following table describes the datatype value.

Table 6-8: REMOTE_DATATYPE value

Bit(s)	Description
Bits 0-7	ODBC (target) datatype (can be extended for types not defined in ODBC)
Bit 8	Returns 1 if nullable, 0 if not nullable
Bit 9	Returns 1 if case sensitive, 0 if not case sensitive
Bits 10, 11	Always returns 10 (binary) meaning updatability unknown
Bits 12, 13	Reserved, always returns 00 (binary)
Bits 14, 15	Returns the following: 01 (binary) meaning NEWODBCDATATYPE (used for all except REAL) 10 (binary) meaning NEWUSERTYPE (used for REAL)
For numeric types:	
Bits 16–23	Precision
Bits 24–31	Scale
For non-numeric types:	
Bits 16–31	Length

sp_databases

Description

Returns a list of databases on a target DBMS.

Syntax

sp_databases

Parameters

This procedure does not allow parameters.

Usage

Information is based on the SYSDATABASE system catalog table.

Results

sp_databases returns a list of databases available to the client. Results are ordered by DATABASE_NAME.

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_databases.

Table 6-9: Result set for sp_databases

Column	Datatype	Description
DATABASE_NAME	varchar(32) NOT NULL	Name of an available database
DATABASE_SIZE	int	Size of the named database in kilobytes, otherwise NULL
REMARKS	varchar(254)	Always NULL

sp_datatype_info

Description

Returns information about a particular datatype or about all supported datatypes.

Syntax

sp_datatype_info [*data_type*]

Parameters

data_type is the ODBC code number for the specified datatype about which sp_datatype_info returns information. See Table 6-7 on page 62 for a description of these codes.

Usage

- The *data_type* parameter specifies the ODBC datatype for which information is requested. If this parameter is not provided, sp_datatype_info returns information about all supported datatypes.
- This function corresponds to the ODBC function SQLGetTypeInfo.

Results

sp_datatype_info returns a list of datatypes with information about each. Results are ordered by the following columns:

- DATA_TYPE
- TYPE_NAME

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database. The following table shows the result set for sp_datatype_info.

Table 6-10: Result set for sp_datatype_info

Column	Datatype	Description
TYPE_NAME	varchar(128) NOT NULL	Name of the Transact-SQL datatype or the target database datatype that corresponds to the ODBC datatype in the DATA_TYPE column.
DATA_TYPE	smallint NOT NULL	ODBC datatype to which all columns of this type are mapped.
PRECISION	int	Maximum precision allowed for this datatype. (NULL is returned for datatypes where precision is not applicable.)
LITERAL_PREFIX	varchar(128)	Character(s) used to prefix a literal; NULL is returned for datatypes where a literal prefix is not applicable.
LITERAL_SUFFIX	varchar(128)	Character(s) used to mark the end of a literal; NULL is returned for datatypes where a literal suffix is not applicable.
CREATE_PARAMS	varchar(128)	Description of the creation parameters required for this datatype, for example, precision and scale; NULL is returned if the datatype does not have creation parameters.
NULLABLE	smallint NOT NULL	Indicates whether the datatype accepts NULL values: • 0 – the column does not accept NULL values. • 1 – the column accepts NULL values.
CASE_SENSITIVE	smallint NOT NULL	Indicates whether the datatype distinguishes between uppercase and lowercase characters: • 0 – the datatype is not a character type or is not case sensitive. • 1 – the datatype is a character type and is case sensitive.
SEARCHABLE	smallint NOT NULL	Indicates how this datatype is used in where clauses: • 0 – the datatype cannot be used in a where clause. • 1 – the datatype can be used in a where clause.
UNSIGNED_ATTRIBUTE	smallint	Indicates whether this attribute is unsigned: • 0 – the datatype is signed. • 1 – the datatype is unsigned. • NULL – the datatype is not numeric.

Column	Datatype	Description
MONEY	smallint NOT NULL	Indicates whether this is a money datatype: • 0 – it is not a money datatype. • 1 – it is a money datatype.
AUTO_INCREMENT	smallint	Indicates whether this datatype automatically increments: • 0 – columns of this datatype do not automatically increment. • 1 – columns of this datatype automatically increment. • NULL – the column is not numeric and does not have a sign.
LOCAL_TYPE_NAME	varchar(128)	The database name or the Transact-SQL® name for the datatype.
MINIMUM_SCALE	smallint	Minimum scale for the datatype; NULL if scale is not applicable.
MAXIMUM_SCALE	smallint	Maximum scale for the datatype; NULL if scale is not applicable.

sp_fkeys

Description

Returns primary and foreign key information for the specified table or tables. Foreign keys must be declared using the ANSI integrity constraint mechanism.

Syntax

sp_fkeys *pktable_name* [, *pktable_owner*]
[, *pktable_qualifier*] [, *fktable_name*]
[, *fktable_owner*] [, *fktable_qualifier*]

Parameters

pktable_name is the name of the table containing the primary key. Views, aliases, and wildcard-character search patterns are not supported. You must specify either this parameter or the fktable_name parameter, or both.

pktable_owner is the owner of the table containing the primary key. Wildcard-character search patterns are not supported. If you do not specify this parameter, sp_fkeys looks first for a table owned by the current user and then for a table owned by the database owner.

pktable_qualifier is ignored. Leave blank or set to NULL.

fktable_name is the name of the table containing the foreign key. Wildcard-character search patterns, views, and aliases are not supported.

fktable_owner is the owner of the table containing the foreign key. Wildcard-character search patterns are not supported. If you do not specify this parameter, sp_fkeys looks first for a table owned by the current user and then for a table owned by the database owner.

fktable_qualifier is ignored. Leave blank or set to NULL.

Usage

- This function corresponds to the ODBC function SQLForeignKeys.
- Information is based on the SYSCOLUMNS, SYSFOREIGNKEYS, SYSINDEXES, SYSRELS, and SYSSYNONYMS system catalog tables.
- For information about creating a foreign key, see the appropriate *IBM DATABASE 2 SQL Reference* manual.

Results

sp_fkeys returns a row for each column that is part of the foreign key or primary key in a primary key/foreign key relationship.

Results are ordered by the following columns:

- PKTABLE_OWNER
- PKTABLE_NAME
- KEY_SEQ

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_fkeys.

Table 6-11: Result set for sp_fkeys

Column	Datatype	Description
PKTABLE_QUALIFIER	varchar(128)	NULL
PKTABLE_OWNER	varchar(128)	Primary key table owner
PKTABLE_NAME	varchar(128) NOT NULL	Primary key table name
PKCOLUMN_NAME	varchar(128) NOT NULL	Primary key column name
FKTABLE_QUALIFIER	varchar(128)	NULL
FKTABLE_OWNER	varchar(128)	Foreign key table owner
FKTABLE_NAME	varchar(128) NOT NULL	Foreign key table name
FKCOLUMN_NAME	varchar(128) NOT NULL	Foreign key column name

Column	Datatype	Description
KEY_SEQ	smallint NOT NULL	Column sequence number in key (starting with 1)
UPDATE_RULE	smallint	Action to be applied to the foreign key when the SQL operation is update: • 0 means cascade. • 1 means restrict. • 2 means set null. • NULL means not applicable to the target database.
DELETE_RULE	smallint	Action to be applied to the foreign key when the SQL operation is delete: • 0 means cascade. • 1 means restrict. • 2 means set null. • NULL means not applicable to the target database.
FK_NAME	varchar(128)	Foreign key identifier; NULL if not applicable to the target database
PK_NAME	varchar(128)	Primary key identifier; NULL if not applicable to the target database

sp_mda

Description

Returns server metadata.

Syntax

sp_mda *parm1*, *parm2*, *parm3*

Parameters

parm1 is a required parameter that indicates the client type. There are three valid values: 0, 1, or 2.

- 0 – returns MDAVERSION and MDARELEASEID rows.
- 1 – returns all the currently defined metadata.
- 2 – returns metadata limited to the following rows:

- MDAVERSION
- CONNECTCONFIG
- SET_CATALOG
- GET_CATALOG
- SET_AUTOCOMMIT_ON
- SET_AUTOCOMMIT_OFF
- SET_ISOLATION
- SET_ROWCOUNT
- OUTERJOINS

parm2 is a required parameter that indicates the client request for sp_mda version to be returned. This parameter is not currently used by SYOSPMDA, and the MDAVERSION metadata is actually returned as a result of *parm1*. The valid values for *parm2* are 0 and 1.

parm3 is an optional parameter that indicates the client version. This parameter is not currently used by SYOSPMDA. The range of valid values is 0-9.

Usage

- sp_mda consists of two programs: SYOSPMDA and SPMDACSP. SYOSPMDA establishes a server connection through the Server Option and reads the three parameters supplied with sp_mda. SPMDACSP determines if the user has chosen to use a DB2 table or a VSAM file as the means to store metadata attributes.
- To execute sp_mda, initiate an isql session to your CICS region, and enter commands like the following:

```
exec sp_mda parm1,parm2,parm3
go
```

Results

The metadata returned consists of three columns:

- MDINFO
- QUERYTYPE
- QUERY

The MDINFO column indicates the metadata attribute name.

The QUERYTYPE column indicates how the QUERY column should be treated. The range of valid values is as follows:

- 1 – Treat QUERY as RPC
- 2 – Treat QUERY as language command
- 3 – Treat QUERY as unsupported
- 4 – Treat QUERY as Boolean literal
- 5 – Treat QUERY as integer literal
- 6 – Treat QUERY as string literal
- 7 – Treat QUERY as integer list literal

The QUERY column is a 255-byte varchar field containing either:

- A literal,

  or

- A query for execution at runtime to retrieve information. For queries that require parameters, a ? (question mark) is used to mark parameters.

The following example illustrates use and return values for sp_mda:

```
1>exec sp_mda 0,3
2>go
```

Table 6-12: Result set for sp_mda

MDINFO	QUERYTYPE	QUERY
MDARELEASED	6	`DirectConnect QTS 220842`
MDAVERSION	5	`7`

sp_pkeys

Description

Returns primary key information for the specified table or tables.

Syntax

sp_pkeys *table_name* [, *table_owner*]
[, *table_qualifier*]

Parameters

table_name is the name of the table. Wildcard-character search patterns, views, and aliases are not supported.

table_owner is the owner of the table. Wildcard-character search patterns are not supported. If you do not specify this parameter, sp_fkeys looks first for a table owned by the current user and then for a table owned by the database owner.

table_qualifier is ignored. Leave blank or set to NULL.

Usage

- This function corresponds to the ODBC function SQLPrimaryKeys.
- Information is based on the SYSINDEXES, SYSKEYS, and SYSSYNONYMS system catalog tables.
- For information about creating a foreign key, see the appropriate *IBM DATABASE 2 SQL Reference*.

Results

sp_pkeys returns a row for each column in the primary key. Results are ordered by:

- TABLE_OWNER
- TABLE_NAME
- KEY_SEQ

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_pkeys.

Table 6-13: Result set for sp_pkeys

Column	Datatype	Description
TABLE_QUALIFIER	varchar(128)	NULL
TABLE_OWNER	varchar(128)	Primary key table owner (authorization ID)
TABLE_NAME	varchar(128) NOT NULL	Primary key table name
COLUMN_NAME	varchar(128) NOT NULL	Primary key column name
KEY_SEQ	smallint NOT NULL	Sequence number of the column in a multi-column primary key
PK_NAME	varchar(128)	Primary key identifier; NULL if not applicable to the target database

sp_server_info

Description

Returns a list of attribute names and matching values for the target DBMS.

Syntax

sp_server_info [*attribute_id*]

Parameters

attribute_id is the integer ID of the attribute. Wildcard-character search patterns are not supported.

Usage

- If the *attribute_id* parameter is not provided, sp_server_info returns information about all attributes.
- This function does not correspond to any ODBC function but returns some of the information returned by SQLGetInfo.

Results

sp_server_info returns a list of the requested attributes and their values.

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_server_info.

Table 6-14: Result set for sp_server_info

Column	Datatype	Description
ATTRIBUTE_ID	int NOT NULL	Numeric identifier of the attribute
ATTRIBUTE_NAME	varchar(60)	Attribute name
ATTRIBUTE_VALUE	varchar(254)	Attribute value

sp_special_columns

Description

Retrieves the following information about columns within a specified table or view:

- The optimal set of columns that uniquely identify a row in the table or view
- A list of the columns that are automatically updated when any value in the row is updated

Syntax

sp_special_columns *table_name* [, *table_owner*]
[, *table_qualifier*] [, *col_type*]

Parameters

table_name is the name of the table. Views, aliases, and wildcard-character search patterns are not supported.

table_owner is the owner of the table. Wildcard-character search patterns are not supported. If you do not specify this parameter, sp_special_columns looks first for a table owned by the current user and then for a table owned by the database owner.

table_qualifier is ignored. Leave blank or set to NULL.

col_type is a value that requests information about columns of a specific type as follows:

- R returns information about columns with values that uniquely identify any row in the table.
- V returns information about columns with values that are automatically generated by a target each time a row is inserted or updated.

Usage

- This function corresponds to the ODBC function SQLSpecialColumns.
- Information is based on the SYSINDEXES, SYSKEYS, and SYSCOLUMNS system catalog tables.

Results

sp_special_columns returns information about the columns that uniquely identify a row in a table.

The result set consists of a row for each column of an index that uniquely identifies each row of the table. If there are multiple unique indexes on a table, the one that is described by the result set is the first that exists in the following list:

- A primary key with clustered index
- A primary key without clustered index
- A unique, clustered index
- A unique, non-clustered index

The result set is ordered by the column name in the index.

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_special_columns.

Table 6-15: Result set for sp_special_columns

Column	Datatype	Description
SCOPE	smallint NOT NULL	Actual scope of the row ID: • 0 SQL_SCOPE_CURROW • 1 SQL_SCOPE_TRANSACTION
COLUMN_NAME	varchar(128) NOT NULL	Column name.
DATA_TYPE	smallint NOT NULL	ODBC datatype to which all columns of this type are mapped.
TYPE_NAME	varchar(128) NOT NULL	Name of the target database datatype that corresponds to the ODBC datatype in the DATA_TYPE column.
PRECISION	int	Maximum precision for the datatype in the target database; NULL if precision is not applicable.
LENGTH	int	Length of the column in bytes.
SCALE	smallint	Number of digits to the right of the decimal point; NULL if scale is not applicable.

Column	Datatype	Description
PSEUDO_COLUMN	smallint	Indicates whether the column is a pseudo-column; the server always returns 0 SQL_PC_UNKNOWN

sp_sproc_columns

Description

Returns descriptive information for the input and return parameters for stored procedures in the current environment.

Syntax

sp_sproc_columns *sp_name* [, *sp_owner*]
[, *sp_qualifier*] [, *column_name*]

Parameters

sp_name is the name of the stored procedure. Use the wildcard character to request information about more than one stored procedure.

sp_owner is the owner of the stored procedure. Use the wildcard character to request information about stored procedures owned by more than one user. If you do not specify this parameter, sp_sproc_columns looks first for a procedure owned by the current user and then for a procedure owned by the database owner.

sp_qualifier is ignored. Leave blank or set to NULL.

column_name is the set of columns to be included in the result set. Use the wildcard character to request information about more than one column. If you do not supply a *column_name* parameter, sp_sproc_columns returns information about all columns for the stored procedure.

Usage

- The server selects information from the SYSPROCCOLUMNS table. The *cspdb2.sql* script creates this table during installation of DirectConnect. However, you need to update the SYSPROCCOLUMNS table manually.
- This function corresponds to the ODBC function SQLProcedureColumns.

Results

sp_sproc_columns returns a list of available procedures. Results are ordered by the following columns:

- PROCEDURE_OWNER

- PROCEDURE_NAME
- COLUMN_TYPE

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_sproc_columns.

Table 6-16: Result set for sp_sproc_columns

Column	Datatype	Description
PROCEDURE_QUALIFIER	varchar(128)	Always NULL
PROCEDURE_OWNER	varchar(128)	Value from the corresponding column of SYSPROCCOLUMNS table
PROCEDURE_NAME	varchar(128) NOT NULL	Name of the stored procedure
COLUMN_NAME	varchar(128) NOT NULL	Name of the input parameter or result set column
COLUMN_TYPE	smallint NOT NULL	Type of data in this procedure column: • 1 SQL_PARAM_INPUT – the procedure column is an input parameter • 3 SQL_RESULT_COL – the procedure column is a result set column
DATA_TYPE	smallint NOT NULL	Integer code for the ODBC SQL datatype equivalent of the target database datatype for this procedure column
TYPE_NAME	varchar(128) NOT NULL	String representing the datatype name in the target database
PRECISION	int	Precision of the procedure column on the target database; NULL if precision is not applicable
LENGTH	int	Length of the column in bytes
SCALE	smallint	Number of digits to the right of the decimal point; NULL if scale is not applicable
RADIX	smallint	Base for numeric types; NULL if radix is not applicable
NULLABLE	smallint	Indicates whether the procedure column accepts NULL values: • 0 – the column does not accept NULL • 1 – the column accepts NULL • 2 – it is not known if the column accepts NULL values
REMARKS	varchar(254)	Description of the procedure column

sp_statistics

Description

Returns statistics information for a single table and the indexes associated with that table.

Syntax

sp_statistics *table_name* [, *table_owner*]
[, *table_qualifier*] [, *index_name*] [, *is_unique*]

Parameters

table_name is name of the table. Wildcard-character search patterns are not supported.

table_owner is the owner of the database object about which column privilege information is requested. Wildcard-character search patterns are not supported. If you do not specify this parameter, sp_statistics looks first for a table owned by the current user and then for a table owned by the database owner.

table_qualifier is ignored. Leave blank or set to NULL.

index_name is the name of the index. Wildcard-character search patterns are not supported.

is_unique is one of the following values:
Y if unique indexes are to be returned
N if unique indexes are not to be returned

Usage

- If *index_name* is specified, sp_statistics returns only information about that index.
- This function corresponds to the ODBC function SQLStatistics.

Results

sp_statistics returns information about the named table. Results are ordered by the following columns:

- NON_UNIQUE
- TYPE
- INDEX_QUALIFIER
- INDEX_NAME

- SEQ_IN_INDEX

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_statistics.

Table 6-17: Result set for sp_statistics

Column	Datatype	Description
TABLE_QUALIFIER	varchar(128)	Always NULL.
TABLE_OWNER	varchar(128)	Table owner authorization ID.
TABLE_NAME	varchar(128) NOT NULL	Name of the table or view.
NON_UNIQUE	smallint	Indicates whether the index permits duplicate values: • 0 (FALSE) means the index prohibits duplicate values • 1 (TRUE) means the index allows duplicate values • NULL is returned if TYPE is SQL_TABLE_STAT
INDEX_QUALIFIER	varchar(128)	Always NULL.
INDEX_NAME	varchar(128)	Index name; NULL is returned if TYPE is SQL_TABLE_STAT.
TYPE	smallint NOT NULL	Type of information returned: • 0 SQL_TABLE_STAT – statistics for a table • 1 SQL_INDEX_CLUSTERED – a clustered index • 2 SQL_INDEX_HASHED – a hashed index • 3 SQL_INDEX_OTHER – another type of index
SEQ_IN_INDEX	smallint	Sequence of the column in the index (the first column is 1); NULL is returned if TYPE is SQL_TABLE_STAT.
COLUMN_NAME	varchar(128)	Column name; NULL is returned if TYPE is SQL_TABLE_STAT.
COLLATION	char(1)	Sort sequence for the column: • A – ascending • D – descending • NULL – returned if TYPE is SQL_TABLE_STAT
CARDINALITY	int	Cardinality of the table or index: • Number of rows in the table if TYPE is SQL_TABLE_STAT • Number of unique values in the index if TYPE is not SQL_TABLE_STAT • NULL if the value is not available from the target database

Column	Datatype	Description
PAGES	int	Number of pages used to store the index or table: • Number of pages used to store the table if TYPE is SQL_TABLE_STAT • Number of pages used to store the index if TYPE is not SQL_TABLE_STAT • NULL if this information is not available from the target database
FILTER_CONDITION	varchar(128)	If the index is a filtered index, this is the filter condition; if the filter condition cannot be determined, this is an empty string. NULL is returned if the index is not a filtered index or TYPE is SQL_TABLE_STAT.

sp_stored_procedures

Description

Returns a list of available procedures.

Syntax

sp_stored_procedures [*sp_name*] [, *sp_owner*]
[, *sp_qualifier*]

Parameters

sp_name is the stored procedure name. Use the wildcard character to request information about more than one stored procedure. If left blank, sp_stored_procedures returns information for all procedures.

sp_owner is the owner of the stored procedure. Use the wildcard character to request information about procedures owned by more than one user.

sp_qualifier is ignored. Leave blank or set to NULL.

Usage

This function corresponds to the ODBC function SQLProcedures.

Results

sp_stored_procedures lists and describes stored procedures. Results are ordered by the following columns:

- *PROCEDURE_QUALIFIER*
- *PROCEDURE_OWNER*

- *PROCEDURE_NAME*

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table describes the result set for sp_stored_procedures.

Table 6-18: Result set for sp_stored_procedures

Column	Datatype	Description
PROCEDURE_QUALIFIER	varchar(128)	Always NULL.
PROCEDURE_OWNER	varchar(128)	Procedure owner.
PROCEDURE_NAME	varchar(128) NOT NULL	Procedure name.
NUM_INPUT_PARAMS	int NOT NULL	Number of input parameters in the stored procedure. -1 – the number of input parameters is unknown.
NUM_OUTPUT_PARAMS	int NOT NULL	Number of return parameters in the stored procedure. -1 – the number of return parameters is unknown.
NUM_RESULT_SETS	int NOT NULL	Number of result sets returned by the stored procedure. -1 – the number of result sets is unknown.
REMARKS	varchar(254)	Describes the procedure.
PROCEDURE_TYPE	smallint	Defines the procedure type: • 0 SQL_PT_UNKNOWN – it cannot be determined whether the procedure returns a value. • 1 SQL_PT_PROCEDURE – the returned object is a procedure; it does not have a return value. • 2 SQL_PT_FUNCTION – the returned object is a function; it has a return value.

sp_table_privileges

Description

Returns privilege information for one or more database objects.

Syntax

sp_table_privileges *table_name* [, *table_owner*]
[, *table_qualifier*]

Parameters

table_name is the name of the table. Use the wildcard character to request information about more than one table. Aliases are not supported.

table_owner is the owner of the database object about which column privilege information is requested. Use the wildcard character to request information about tables owned by more than one user. If you do not specify this parameter, sp_table_privileges looks first for a table owned by the current user and then for a table owned by the database owner.

table_qualifier is ignored. Leave blank or set to NULL.

Usage

- The server selects information from the SYSTABAUTH system catalog table.
- This function corresponds to the ODBC function SQLTablePrivileges.

Results

sp_table privileges returns a list of one or more database objects with privilege information about each. Results are ordered by the following columns:

- TABLE_OWNER
- TABLE_NAME
- PRIVILEGE

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_table_privileges.

Table 6-19: Result set for sp_table_privileges

Column Name	Datatype	Notes
TABLE_QUALIFIER	varchar (128)	Always NULL
TABLE_OWNER	varchar (128)	Table owner identifier (authorization ID)
TABLE_NAME	varchar (128) NOT NULL	Name of the database object about which privilege information is returned
GRANTOR	varchar (128)	Identifies the user who granted this privilege; NULL if not applicable to the target database
GRANTEE	varchar (128) NOT NULL	Identifies the user to whom this privilege was granted

Column Name	Datatype	Notes
PRIVILEGE	varchar (128) NOT NULL	Identifies the privilege granted to the grantee on this object as one of the following values: • SELECT – the grantee is authorized to select rows in the associated object. • INSERT – the grantee is authorized to insert rows into the associated object. • UPDATE – the grantee is authorized to update rows in the associated object. • REFERENCES – the grantee is authorized to refer to one or more columns of the table within a constraint (for example: unique, referential, or table check constraint).
IS_GRANTABLE	varchar (3)	Indicates whether the grantee is authorized to grant privilege on this object to other users with one of the following values: • YES – the grantee can grant this privilege to others. • NO – the grantee cannot grant this privilege to others. • NULL – it is unknown or not applicable to the target database.

sp_tables

Description

Returns a list of objects stored in the database.

Syntax

sp_tables [*table_name*] [, *table_owner*]
[, *table_qualifier*] [, *table_type*]

Parameters

table_name is the name of the table. Use the wildcard character to request information about more than one table.

table_owner is the owner of the table. Use the wildcard character to request information about tables owned by more than one user.

table_qualifier is ignored. Leave empty or set to NULL.

table_type is a list of values, separated by commas, requesting information about all objects of a specific type(s) as follows:

"'TABLE', 'SYSTEM TABLE', 'VIEW', 'ALIAS', 'SYNONYM'"

Note You must enclose each table type with single quotation marks, and enclose the entire parameter with double quotation marks. Enter table types in uppercase.

Usage

This function corresponds to the ODBC function SQLTables.

Results

sp_tables returns a list of database objects. Results are ordered by the following columns:

- *TABLE_TYPE*
- *TABLE_OWNER*
- *TABLE_NAME*

The lengths for varchar columns shown in the result set tables are maximums; the actual lengths depend on the target database.

The following table shows the result set for sp_tables.

Table 6-20: Result set for sp_tables

Column	Datatype	Description
TABLE_QUALIFIER	varchar(128)	Always NULL
TABLE_OWNER	varchar(128)	Table owner
TABLE_NAME	varchar(128)	Name of the object about which information is returned
TABLE_TYPE	varchar(128) NOT NULL	One of the following: • 'ALIAS' • 'SYNONYM' • 'SYSTEM TABLE' • 'TABLE' • 'VIEW'
REMARKS	varchar(254)	A description of the table or NULL

sp_thread_props

Description

Allows the client to retrieve and set various thread properties.

Syntax

sp_thread_props [*property_name* [, *property_value*]]

Parameters

property_name is the name of the property to be set or shown.

property_value is the value to which the property is to be set.

Usage

If you do not provide any parameters, or if you provide only *property_name*, the server returns a single result set consisting of every instance of *property_name* and the value for each.

Using CSPs to access non-catalog DB2 information

The DB2 database administrator may need to access Sybase-defined tables that store requested CSP information not available in the DB2 catalog.

Note For comparable Access Service information, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*.

This section discusses CSPs and the DBMS accessed.

CSPs and DBMS

Several CSPs access tables that the DB2 database administrator maintains, including the tables shown in the following table.

Table 6-21: DB2 Database administrator-maintained tables

Member	Description
sp_server_info	The server, including capabilities and limitations
sp_sproc_columns	Stored procedures that an application can execute
sp_stored_procedures	Stored procedures executable from DirectConnect

During installation of the CSPs, the tables that these CSPs query are created and populated.

For details on the CSPs, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*.

CSPs in a non-DB2 environment

If DB2 is not available in the CICS region, rerun the RPC install task from the Server Option for CICS install.

CHAPTER 7 User Exits

This chapter describes how to use the DB2 UDB Option for CICS exits and provides examples of the kinds of programs you can write.

Note Be sure you compile and link-edit the DB2 UDB Option for CICS sample programs so that they run in 31-bit addressing mode. The samples are in the *SOURCE* library.

This chapter covers the following topics:

User exits in the DB2 UDB Option for CICS

User exits enable you to control the flow of execution in the DB2 UDB Option for CICS and to control incoming and outgoing data. The DB2 UDB Option for CICS provides a communications area in which to write applications to:

- Validate requests and results processed by the DB2 UDB Option for CICS
- Generate a list of group IDs for input to Catalog Stored Procedures (CSPs) for DB2 AS only
- Customize SQL commands for a particular database or a specific purpose

Note For information about providing security, see the Mainframe Connect Server Option for CICS *Installation and Administration Guide*.

Request/Result validation exits

When the DB2 UDB Option for CICS passes a request to DB2 for processing, AMD2CICS can invoke a request/result validation exit. When AMD2CICS invokes a request/validation exit, the following events occur:

1. AMD2CICS uses the standard EXEC CICS LINK command with a common area described in the copy library book AMD2CCPX.
2. The request validation exit checks whether the request is valid by using values in the Exit Communication Area (described in "Understanding the Exit Communication Area" on page 91).
3. After the request processes, AMD2CICS calls a result exit. AMD2CICS uses the standard EXEC CICS LINK with a common area described in the EXITAREA assembler macro to validate the results and determine whether to give control back to the DB2 UDB Option for CICS. AMD2CICS also uses values in the Exit Communication Area to determine whether to give control back to the DB2 UDB Option for CICS.

The DB2 UDB Option for CICS installation contains examples of validation exit programs.

This section explains how to write a request or request validation exit and contains the following subsections:

- Understanding exit routines
- Enabling request/result validation exits
- Using the sample validation program
- Understanding the Exit Communication Area
- Writing a request validation exit
- Writing a result validation exit
- Using the Group ID Exit Name
- Understanding Group ID Exit Name processing

Understanding exit routines

At appropriate points in the request processing sequence, the DB2 UDB Option for CICS issues calls to your exit program. You can write an exit program to:

- Enforce special installation security rules

- Prohibit access at certain times
- Capture statistics and user information
- Examine or modify the user’s SQL statement
- Direct results to other destinations

The DB2 UDB Option for CICS *INSTALL* library includes a sample validation exit program, AMD2PXIT. You can modify this program or create a new one.

Enabling request/result validation exits

You can use the DB2 UDB Option Configuration screen to enable or disable the request and result validation exits. For instructions on accessing the DB2 UDB Option Configuration screen, see “Changing the DB2 UDB Option for CICS configuration” on page 27.

Note If you run a remote stored procedure (RSP), the result validation exit is not called. If you run a USE REQUEST statement, the request validation exit is called once for the USE REQUEST statement, and once for the SQL statement retrieved from the Host Request Library.

Be sure to add an entry to the CICS PPT for each exit program. For details, see “Result exit name” on page 29.

Using the sample validation program

The *SOURCE* library includes a sample validation program AMD2PXIT. The AMD2PXIT program captures user information and puts it in a transient data queue called USERINFO. You can modify this program or create a new one.

```
TITLE 'EXITSAMP  SAMPLE EXIT PROCESSING'
 PRINT NOGEN
 *---------------------------------------------------------------------*
 * THIS SAMPLE EXIT PROGRAM IS CALLED FOR BOTH REQUEST AND RESULT      *
 * EXITS.  FOR THE REQUEST EXIT, THE REQUEST IS LOGGED TO A TEMPORARY  *
 * STORAGE QUEUE CALLED 'USERINFO'.  IN ADDITION, IF THE REQUEST NAME  *
 * IS 'PAYROLL', THE REQUEST IS IGNORED.  FOR THE RESULT EXIT, THE     *
 * SIZE OF REPORT1 IS CHECKED.                                         *
 *---------------------------------------------------------------------*
 R0       EQU   0
```

```
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
BLANK    EQU   X'40'
EJECT
*------------------*
*  INITIALIZATION  *
*------------------*
EXITSAMP DFHEIENT CODEREG=(12),DATAREG=(13)
USING EXITAREA,R8             ESTABLISH DSECT ADDRESSABILITY
L     R8,DFHEICAP             GET ADDRESS OF EXITAREA
*-------------------------------------*
*  HANDLE A CALL TO THE REQUEST EXIT  *
*-------------------------------------*
REQUEST  CLC   XREASON,=CL8'REQUEST'    CALL TO REQUEST EXIT ?
BNE   RESULT                  NO, BRANCH
CLC   XREQNAME,=CL8'PAYROLL'   IS THIS A PAYROLL REQUEST ?
BE    IGNORE                  YES, IGNORE THE REQUEST
MVC   LOGREC,BLANKS           CLEAR LOG RECORD
MVC   LOGREC+00(08),XDATE      DATE
MVC   LOGREC+09(08),XTIME      TIME
MVC   LOGREC+18(08),XUSERID    USERID
MVC   LOGREC+27(08),XREQNAME   REQUEST NAME
MVC   LOGREC+36(08),XREQFUNC   REQUEST FUNCTION
EXEC  CICS WRITEQ TS QUEUE('USERINFO')                       +
FROM(LOGREC) LENGTH(44)
B     RETURN                  RETURN
*------------------------------------*
*  HANDLE A CALL TO THE RESULT EXIT  *
*------------------------------------*
RESULT   CLC   XRESNAME,=CL8'REPORT1'   IS THIS REPORT1 ?
BNE   RETURN                  NO, BRANCH
L     R3,XRESROWS             GET RESULT ROW COUNT ?
C     R3,=F'3000'                IS REPORT TOO LARGE ?
BNH   RETURN                  NO, BRANCH
```

```
MVC   XERRMSG,=CL70'REPORT TOO LARGE.  CHANGE WHERE CLAUSE'
B     IGNORE
*---------------------------------*
*  RETURN TO THE CALLING PROGRAM  *
*---------------------------------*
IGNORE   MVC   XACTION,=CL8'IGNORE'     INDICATE IGNORE
RETURN   DS    0H
EXEC  CICS RETURN
EJECT
*-------------------------*
*  MISCELLANEOUS STORAGE  *
*-------------------------*
BLANKS   DC    CL44' '
LTORG
DFHEISTG
LOGREC   DS    CL44
COPY  EXITAREA
END
```

Understanding the Exit Communication Area

When a call is made to the request or result validation exit, the Exit Communication Area passes information to the exit program.

The Exit Communication Area contains information that is passed to either a request or result validation exit program when a call is made to that program.

The *MACRO* library includes a description of the Exit Communication Area (EXITAREA), which follows:

```
***********************************************************************
*  COMMUNICATION AREA PASSED TO THE REQUEST/RESULT EXITS              *
***********************************************************************
EXITAREA DSECT
XHEADER  DS    CL8                      EYE CATCHER
XREASON  DS    CL8                      REASON FOR CALLING EXIT
XUSERID  DS    CL8                      USER IDENTIFICATION
XTERMID  DS    CL4                      TERMINAL IDENTIFICATION
XDATE    DS    CL8                      DATE IN MM/DD/YY FORMAT
XTIME    DS    CL8                      TIME IN HH:MM:SS FORMAT
XSERVER  DS    CL8                      NAME OF SERVER
XDBSID   DS    CL30                     DATABASE SYSTEM ID
XDBUID   DS    CL30                     DATABASE USER ID
*----------------------------*
```

```
*  REQUEST EXIT INFORMATION  *
*----------------------------*
XREQINFO DS    0CL28                    REQUEST INFORMATION
XREQNAME DS    CL8                      REQUEST NAME
XREQADDR DS    F                        ADDRESS OF REQUEST BUFFER
XREQFUNC DS    CL8                      REQUEST FUNCTION
XREQOPT  DS    CL8                      REQUEST OPTION
*----------------------------*
*  RESULT EXIT INFORMATION   *
*----------------------------*
XRESINFO DS    0CL36                    RESULT INFORMATION
XRESNAME DS    CL8                      RESULT NAME
XRESOPT  DS    CL8                      RESULT OPTION
XRESTYPE DS    CL3                      RESULT TYPE
XRESFILL DS    CL1                      FILLER
XRESCODE DS    F                        DB2 RETURN CODE
XRESROWS DS    F                        NO. OF ROWS IN RESULT
XRESSIZE DS    F                        NO. OF BYTES IN RESULT
XRESQNAM DS    CL8                      TEMPORARY STORAGE QUEUE NAME
*---------------------------------------------------*
*  VALUES THAT CAN BE MODIFIED BY THE EXIT PROGRAM  *
*---------------------------------------------------*
XACTION  DS    CL8                      ACTION TO BE TAKEN
XDDBNAME DS    CL8                      DEFAULT DATABASE NAME
XDTSNAME DS    CL8                      DEFAULT TABLESPACE NAME
XERRMSG  DS    CL70                     ERROR MESSAGE
XFILLER  DS    CL30                     FILLER
```

Common fields

When the DB2 UDB Option for CICS invokes the exit program, it always passes certain fields from the Exit Communication Area to the exit program, regardless of the validation exit that was called. These fields are as follows:

Table 7-1: Common Exit Communication Area fields

Field	Description
XHEADER	A header containing the word EXITAREA.
XREASON	The reason for calling the exit. REQUEST indicates a call to the request validation exit. RESULT indicates a call to the result validation exit.
XUSERID	The user identification.
XTERMID	The terminal identification.
XDATE	The current date in the format *MM*/*DD*/*YY*.
XTIME	The current time of day in the format *HH*:*MM*:*SS*.
XSERVER	The database server name.
XDBSID	The database system ID, which is not applicable for DB2.
XDBUID	The database user ID, which is not applicable for DB2.

The values for these fields are shown in the example in “Understanding the Exit Communication Area” on page 91, following the first comment area.

Writing a request validation exit

The DB2 UDB Option for CICS calls the request validation exit before a request is executed by the DB2 UDB Option for CICS. The exit program examines the Exit Communication Area passed by the DB2 UDB Option for CICS. Based on the information in the XACTION field description, the exit program either passes, fails, or modifies the request.

The request validation exit can be either a separate program or combined with the result validation program. You can check the XREASON field to determine the reason for the call.

See the following section, “Writing a result validation exit,” for details on writing a result validation exit.

The communication area fields that pertain to the request validation exit are as follows:

Table 7-2: Request Validation Communication Area fields

Field	Description
XREASON	A header containing the word REQUEST.
XREQNAME	The name of the request.
XREQFUNC	The request function. Valid values are SELECT, UPDATE, USE, and so on, which are taken from the SQL statement. The first word of the SQL statement is not used anymore.
XREQOPT	The request option. Valid values are PREVIEW and REPORT, which are taken from the SQL statement.
XREQADDR	The address of the request buffer. The request buffer is formatted as a half-word length field followed by the request.
XACTION	This field is blank when the DB2 UDB Option for CICS sends it on to DB2. If you specify it, the exit program can place a value in this field to indicate what action the DB2 UDB Option for CICS takes: • If you leave this field blank, the DB2 UDB Option for CICS does nothing. Processing proceeds in the normal fashion. • If you specify IGNORE, the DB2 UDB Option for CICS ignores the request, and processing continues with the next request in the request queue. This request is not treated as an error condition. • If you specify REJECT, the DB2 UDB Option for CICS rejects the request. This request is treated as an error condition. If a request is ignored or rejected, the DB2 UDB Option for CICS issues an appropriate error message when the results return to the client application. If you specify it, the exit program can supply an additional error message in the XERRMSG field.
XERRMSG	This field is blank.

The values for the request validation exit fields are shown throughout the example in “Understanding the Exit Communication Area” on page 91.

Writing a result validation exit

A corresponding result validation exit runs after all the requested results have been written to temporary storage. The result validation exit deals with the result as a whole, not row-by-row.

The exit program can examine the Exit Communication Area that the DB2 UDB Option for CICS passes to it. Based on the information in the communication area, the exit program can either pass or fail the result.

The communication area fields that pertain to the result validation exit are as follows:

Table 7-3: Result validation communication area fields

Field	Description
XREASON	A header containing the word RESULT.
XRESNAME	The name of the result.
XRESOPT	The result option. The valid value is PREVIEW.
XRESTYPE	The result type. The valid value is OUT.
XRESCODE	The DB2 return code.
XRESROWS	The size of the result in rows.
XRESSIZE	The size of the result in bytes.
XRESQNAM	The queue name of the result file.
XACTION	This field is blank when the DB2 UDB Option for CICS sends it on. The exit program can place a value in this field to indicate which of the following actions the DB2 UDB Option for CICS takes: • If you leave this field blank, the DB2 UDB Option for CICS does nothing. Processing proceeds in the normal fashion. • If you specify IGNORE or REJECT, the DB2 UDB Option for CICS bypasses this result, which means the result exit is called once per result, not once for each row. It does not return the result to the client application. The results might also be rejected if a row count is exceeded.
XERRMSG	The error message or blank. If you want an additional error message in the XERRMSG field, the exit program can supply it.

Note If you use a result exit, set Temp storage queue type to AUX or MAIN at the DB2 UDB Option Configuration panel (see “Temp storage queue type”). The result exit is not called when Temp storage queue type is set to NONE.

The values for the result validation exit fields are shown throughout the example in “Understanding the Exit Communication Area” on page 91.

Using the Group ID Exit Name

The Group ID Exit Name is a CICS program that builds a list of group IDs for CSPs to use with DB2 Access Services.

If the catalog access parameters are properly configured, the Group ID Exit Name can be called before any CSPs are generated. The Group ID Exit Name provides the appropriate list of group IDs used to determine table and column authorizations for CSP results.

CSP exit programs are typically used in the absence of facilities like RACF, ACF2, or Top Secret, but where the equivalent of Group-ID Support is desired. However, if you use security packages like RACF, ACF2, or Top Secret, you can write an additional exit program to limit the result set returned from CSPs, for example, where user IDs belong to large numbers of groups.

The Group ID Exit Name is a dynamic link. To use it, you must add an entry for a user CSP exit program to the CICS PPT. Also, the program must be in the CICS RPL.

Understanding Group ID Exit Name processing

When the Group ID Exit Name is called, the DB2 UDB Option for CICS passes the exit program a 4012-byte communication area into which the exit program can build a list of up to 500 8-byte group IDs.

The format of the communication area is as follows:

```
COMMAREA   DSECT
 COUNT     DS      F
 USERID    DS      CL8
 ENTRIES   DS      500CL8
```

The COUNT field contains the number of entries in the list including the original user ID. ENTRIES can contain a list of 500 8-byte group IDs.

The communication area comes with an entry for USERID already specified. The COUNT field contains 1. The exit program can perform one of the following actions:

- Do nothing. The user ID is used for CSPs.
- Replace the user ID with a new ID. The new ID is used for CSPs.
- Add additional IDs to the list. The additional IDs must follow the entry for user ID, and the COUNT field must change to reflect the total number of IDs in the list (including the user ID). All the IDs are used for the CSPs.

In the example that follows, two group IDs (GROUP1 and GROUP2) are added to the list:

```
EXITPROG  DFHEIENT
          USING   COMMAREA,R2
          L       R2,DFHEICAP
          MVC     COUNT,=F'3'
          MVC     ENTRIES+08(08),=CL8'GROUP1'
          MVC     ENTRIES+16(08),=CL8'GROUP2'
          EXEC CICS RETURN
          LTORG
          DFHEISTG
 COMMAREA DSECT
 COUNT    DS      F
 ENTRIES  DS      501CL8
       END
```

Parse exit routine

The parse exit routine is similar to a request exit. It allows a customer-written exit routine to alter SQL text received from the DB2 UDB Option for CICS clients before the SQL statements are passed to DB2. By altering the SQL text, your routine can customize SQL commands for a particular database or a specific purpose.

There are several potential uses for this option. For example, you can use a parse exit routine to limit the scope of client access by restricting client SQL statements to select statements only. You can also use the parse exit routine to modify the qualifiers for DB2 object names to provide additional security restraints on accessing mainframe data.

This section explains how to write a parse exit routine. It is organized around the following tasks, which are performed by the programmer:

- Understanding the parse exit routine
- Understanding the parse exit common area
- Preparing to code your routine
- Coding your routine
- Compiling and link-editing your routine
- Enabling your routine

A section containing the definitions for the fields that pass values to and from your routine is included in “CICS common area fields” on page 102.

Understanding the parse exit routine

When the CICS transaction AMD2 receives SQL text from a DirectConnect client, it processes the statements and prepares to send the SQL text to DB2. If the parse exit is specified on the DB2 UDB Option Configuration screen, AMD2 calls the parse exit routine before passing the SQL text to DB2. For more information, see “Parse exit name.”

Calling the parse exit from AMD2

For CICS, AMD2 uses the EXEC CICS LINK command to call the parse exit routine. The call is executed only if the parse exit option is selected using the DB2 UDB Option Configuration screen. See “Changing the DB2 UDB Option for CICS configuration” for instructions on using the DB2 UDB Option Configuration screen.

This option can be selected during installation of the product (see step 3 in “Changing the DB2 UDB Option for CICS configuration” on page 27), or when you enable your routine following the instructions in this section.

Before calling the parse exit routine, AMD2 sets up the parse exit common area that a parse exit routine can access. Your routine should alter the values in the common area of the parse exit as shown in the following table. An explanation of the fields and information about changing their values is included in “CICS common area fields” on page 102.

Table 7-4: Altering values in common area fields

Common area field	When to alter the value
USERID	Never
PASSWORD	Never
NETGWNAME	Never
STMT-PTR	When the SQL text length changes
RETCODE	Always (every invocation of the routine)
MSGCODE	When the return code equals +2

Processing batched SQL statements

AMD2 calls the parse exit routine once for each SQL statement. If the SQL statements are batched using the semicolon delimiter, AMD2 processes each statement separately, calling the parse exit routine and passing the returned SQL text to DB2. Processing continues until one of the following conditions occurs:

- The batch is exhausted.
- The parse exit routine requests that the batch be aborted.
- A DB2 error occurs in processing any statement in the batch.

Processing user-defined transactions

A user-defined transaction is a unit of work explicitly defined by the client application with transaction control statements such as begin tran, commit, and rollback. A unit of work is the data that the users want to confirm was added to the back end database.

The parse exit routine is called once for each SQL statement, including the transaction control statements that define the transaction. Your routine should not alter transaction control statements.

Warning! A customer-defined transaction causes AMD2 to maintain database locks and the DirectConnect-to-mainframe connection until you end the transaction. If your parse exit routine alters the transaction control statements, especially those that explicitly end the transaction, your routine can impact DB2 and other mainframe resource usage.

Returning to AMD2

After customizing the SQL text, the exit routine returns to AMD2, which passes the altered SQL statements to DB2 without further modification.

Understanding the parse exit common area

The parse exit common area passes information between AMD2 and the parse exit routine. It is acquired and initialized by AMD2. A COBOL II copybook, AMD2CCPX, defines the common area fields that are passed (through the linkage section) to a COBOL II parse exit routine. This copybook is distributed with the DB2 UDB Option for CICS.

The layout of the common area, with descriptions of each field, follows.

Layout of the AMD2CCPX copybook

The CICS common area has the following structure. ("DB2 Access Module" in the following copybook means "AMD2.")

```
**************************************************************
*                  AMD2CCPX                       Rel 10.1
* This copy book defines the LINKAGE SECTION for a COBOL II
* Parse Exit Routine for DB2 Access Module.
**************************************************************
       01  AMD2CCPX-COMMAREA           REDEFINES DFHCOMMAREA.
           05  AMD2CCPX-CA-EYECATCHER  PIC X(8).
           05  AMD2CCPX-CA-USERID      PIC X(32).
           05  AMD2CCPX-CA-PASSWORD    PIC X(32).
           05  AMD2CCPX-CA-NETGWNAME   PIC X(32).
           05  AMD2CCPX-CA-CHARSETCLT  PIC X(32).
           05  AMD2CCPX-CA-CHARSETSRV  PIC X(32).
           05  AMD2CCPX-CA-NATLANGCLT  PIC X(32).
           05  AMD2CCPXAMD2CCPX-CA-DBCS PIC X(32).
           05  AMD2CCPX-CA-STMT-PTR POINTER.
           05  AMD2CCPX-CA-RETCODE     PIC S9(9)  COMP.
           05  AMD2CCPX-CA-MSGCODE     PIC S9(9)  COMP.
           05  AMD2CCPX-CA-RESERVED    PIC X(80).
*------------------------------------------------------------
* Define SQL text area pointed to by AMD2CCPX-CA-STMT-PTR
*------------------------------------------------------------
       01  AMD2SQL-TEXT-LVL-01.
           03  AMD2SQL-TEXT.
               05  AMD2SQL-TEXT-LL     PIC S9(4)  COMP.
               05  AMD2SQL-TEXT-CHARS OCCURS 0 TO 32704 TIMES
```

```
                   DEPENDING ON AMD2SQL-TEXT-LL
  PIC X.
           03  AMD2SQL-TEXT-DUMMY-LVL PIC X.
*****************************************************************
* END AMD2CCPX COPY BOOK
*****************************************************************
```

CICS common area fields

The following AMD2CCPX-CA fields are defined for COBOL/CICS.

Table 7-5: CICS common area fields

Field	Description
EYECATCHER	This value is set by AMD2. If it is altered by the parse exit routine, unpredictable results occur.
USERID	The user ID of the client sending the request. This field should not be altered by the parse exit routine.
PASSWORD	The password of the client sending the request. This field should not be altered by the parse exit routine.
NETGWNAME	The name of DirectConnect specified by the client. These fields should not be altered by the parse exit routine.
CHARSETCLT	Reserved for future use.
CHARSETSRV	Reserved for future use.
ATLANGCLT	Reserved for future use.
DBCS	Reserved for future use.
STMT-PTR	A pointer to a single SQL statement beginning with the 2-byte SQL length. The parse exit routine should alter the pointer to hold the new address of the SQL text area when this address is changed by the exit routine. It is only necessary to acquire a new SQL text area if the existing area is not large enough for the SQL text after it is altered by the user. AMD2 uses this pointer to determine whether the exit routine allocated storage that AMD2 must deallocate.
RETCODE	A full-word return code. The user-written parse exit routine should alter this field. Valid values are: 0—Passes the SQL text to DB2. +1—Does not pass the SQL text to DB2 and continues with the next statement in the batch, if there is one. +2—Aborts the batch and returns an error message using the message code in the next parse Exit Communication Area field.
MSGCODE	A fullword message code. This field should be altered by the user-written parse exit routine if the return code was set to +2. AMD2 ignores it in all other cases.
RESERVED	Reserved for future use.

Preparing to code your routine

This section describes your programming language options and the materials that Sybase provides, such as the sample routine and the COBOL copybook, that can help you write the routine.

Programming languages

The parse exit feature is designed to work with assembler, COBOL, and PL/I.

Sybase supplies a sample parse exit routine in COBOL and a COBOL copybook that defines the linkage section for a COBOL II parse exit routine. This routine is in the SOURCE library.

Sample parse exit routine

The sample parse exit routine, AMD2CCPX, is located in the SOURCE library. You can modify this program or use it as an example. The COBOL copybook is discussed in the "Understanding the parse exit common area" on page 100.

Coding your routine

This section contains information about the tasks that you should code your routine to perform. These tasks include:

- Altering the SQL text to suit your purposes
- Acquiring additional storage, if necessary
- Handling errors
- Assigning a return code
- Selecting an error message

Altering the SQL text

The parse exit common area contains a pointer to the SQL text area. The text area consists of the SQL text preceded by a 2-byte SQL length field, which provides the number of bytes in the SQL text. (The SQL length field does not include the 2 bytes for the length field itself.)

If your exit routine changes the length of the SQL text, it must also change the value in the 2-byte length field.

- If the SQL text is shortened by the processing of your exit routine, you need only to correct the value in the length field.
- If the SQL text is expanded by your exit routine, you must acquire additional storage to handle the overflow.

Note As you code your exit routine to alter the SQL text, be aware that AMD2 does not modify the SQL text after your routine is executed. Therefore, the SQL statements must be in the DB2 syntax before the parse exit routine returns to AMD2.

Acquiring additional storage

If your routine expands the SQL statements beyond the length specified in the 2-byte SQL length field, the parse exit routine must acquire additional storage before it alters the SQL text.

Warning! A storage violation can occur if your routine expands the SQL text area beyond the specified length in the 2-byte SQL length field without first acquiring additional storage.

To acquire additional storage for a larger SQL text area, your routine must issue a GETMAIN command. Like the old SQL text area, the new one must allow 2 bytes for the length field in addition to the area needed for the expanded SQL statement.

After altering the new SQL text, your routine must:

- Load the length field with the new length of the SQL text after the field has been expanded
- Change the pointer in the parse exit common area to the location of the new SQL text area

This pointer should be altered by the parse exit routine to hold the new address of the SQL text area if this address is changed by the exit routine. It is only necessary to acquire a new SQL text area if the existing area is not large enough for the SQL text after it is altered by the user.

Your exit routine does not need to deallocate the storage it acquires. After the routine returns, AMD2 checks the SQL text pointer. If the pointer changed during the exit routine execution, AMD2 deallocates the additional storage after DB2 processes the SQL statement.

Handling errors

Errors associated with parse exit routine execution can occur before the routine is called, during the routine's processing, and after the routine returns to AMD2. This section describes the types of errors and their consequences, so that you can provide appropriate error handling code in your routine.

Before the routine is called

The parse exit routine is not invoked if an error occurs in AMD2 processing before the parse exit call. Error 33233 is sent to the client if this occurs.

During parse exit routine processing

Errors that occur during the processing of your routine produce the following results:

- If your routine abnormally ends (abends) and your code does not handle the abend, the DB2 UDB Option for CICS abends and the appropriate messages may not be returned.
- If your routine abends and your code handles the abend, the DB2 UDB Option for CICS may not be aware of the abend, depending upon how your abend handler is written.
- If your routine contains an internal logic error, the results may be unpredictable.

After returning to AMD2

Errors that result from exit routine processing can occur after the routine returns to AMD2.

- If your routine acquires additional storage and incorrectly alters the pointer to the SQL text, the DB2 UDB Option for CICS abends when AMD2 issues a FREEMAIN command to deallocate the additional storage.
- If the modifications of the SQL text are invalid, DB2 performs its usual error handling. Any DB2 error that occurs in batched SQL commands terminates the entire batch, unless the Stop_on_error property is set.

Assigning a return code

Your exit routine must pass a return code back to AMD2 in the parse exit common area field AMD2CCPX-CA-RETCOD. Valid values are:

0—Passes the SQL text on to DB2.

+1—Does not pass the SQL text to DB2 and continues with the next statement, if there is one.

+2—Aborts the batch and returns an error message using the message code in the AMD2CCPX-CA-MSCODE field.

Selecting an error message

If your exit routine sets the return code to +2, it should also change the parse exit common area message code field AMD2CCPX-CA-MSGCODE.

After a parse exit routine returns to AMD2 with a return code of +2, AMD2 matches the message code against the message table AMD2CAMX. If the message code is in the table, AMD2 sends the text associated with that message number. If AMD2 cannot find the message code in the table, it sends the default error message: Error 33231.

Any message code between 33400 and 33999 is available for use as a user message code. AMD2CAMX is shipped with a sample error message (Error 33400), which is coded for the parse exit routine.

Altering the message table

To add messages for your routine to the message table, add entries to the SOURCE library member AMD2CAMX. After making any changes, you must reassemble and relink the message table. The JCL for this job is in JCL Library members AMD2CAMX.

For more information, see “Changing the AMD2 message table.”

Compiling and link-editing your routine

The JCL for compiling and link-editing a parse exit routine is in the JCL library member SYSAPARS. Change the property values to those for your routine before running the job.

Enabling your routine

You can enable your routine by specifying the parse exit name on the DB2 UDB Option Configuration screen, which shows you whether or not it is successfully enabled.

CHAPTER 8 Troubleshooting

This chapter provides troubleshooting guidelines for the mainframe, including instructions for locating errors, descriptions of common problems, and process flow overviews. The information in this chapter applies to both environments using SNA or TCP/IP, except where only one protocol is indicated.

This chapter covers the following topics:

Note For information on troubleshooting DirectConnect, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services* and the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*.

Components and functionality

This section describes the components and functionality that should be considered when troubleshooting problems in the DB2 UDB Option for CICS:

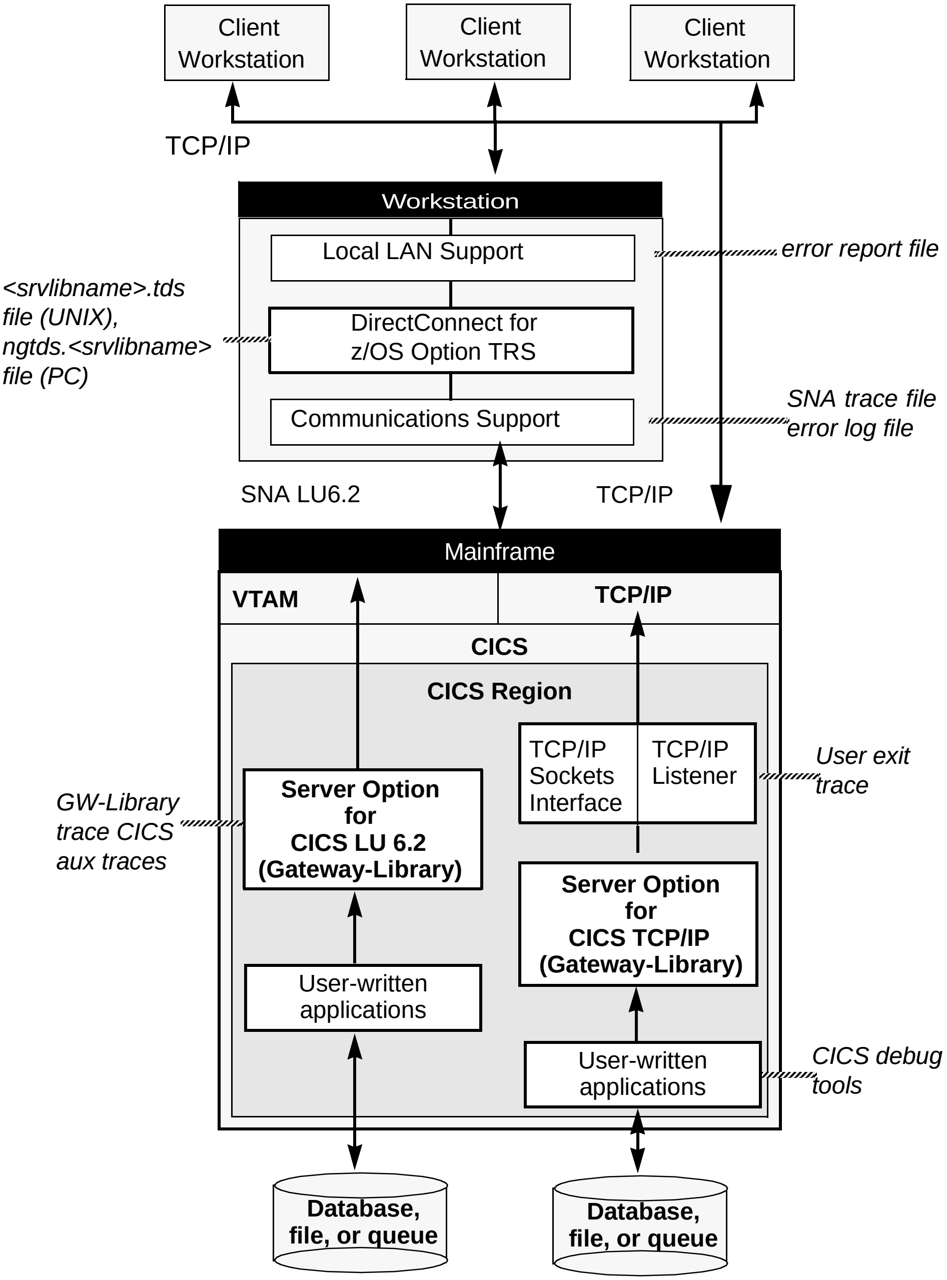
- System components
- Process flow

System components

The DB2 UDB Option for CICS support consists of several components on the IBM z/Series mainframe and the DirectConnect for z/OS Option platform, as the following diagram shows. On the z/OS Option platform, support is aided by information from LAN error report files, TDS and SNA trace files, and connectivity error log files. On the IBM z/Series mainframe, Server Option GW-Library and CICS aux traces, TCP/IP listener traces, and CICS debug tools help in troubleshooting.

The following figure shows the Server Option and DirectConnect for z/OS Option support components in a CICS environment:

Figure 8-1: Server Option and DirectConnect for z/OS Option support components



Note The DB2 UDB Option for CICS is a Server Option application using the Server Option API. The DB2 UDB Option for CICS requires the Server Option to use LU 6.2 and TCP/IP connectivity.

DirectConnect support

DirectConnect performs the following functions:

- Receives requests from client applications
- Converts the requests to the appropriate communications protocol call
- Sends the requests to the mainframe

Each TRS or access service includes a unique server name, which clients use to select a server for communication, and has its own set of configuration information.

As shown in Figure 8-1, DirectConnect uses four files:

- *srv.log* for logging TDS traffic between TRS and client workstations, and for recording errors
- *ngtds* for tracing Sybase TDS traffic between TRS and mainframe SNA
- *servername.log* for logging TDS traffic between the access service and client workstations, and for recording errors
- *tds.trc* for tracing Sybase TDS traffic between the access service and mainframe SNA

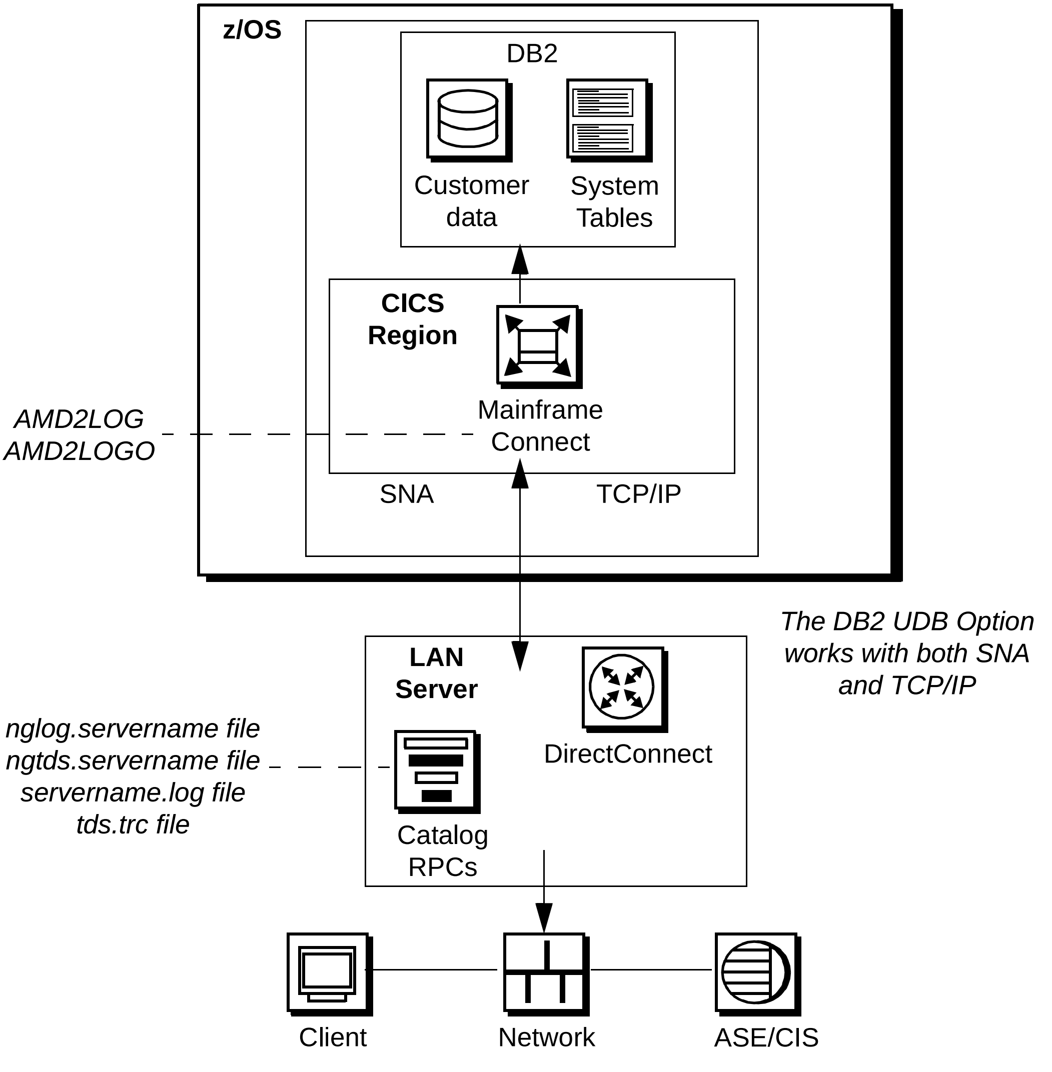
For more information, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services* and the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*.

DB2 UDB Option for CICS support

The DB2 UDB Option for CICS provides the AMD2 transaction, which automatically processes client SQL language requests using the DB2 dynamic SQL facilities.

Using this product, client applications can communicate directly with DirectConnect or with another server that communicates with DirectConnect, such as Adaptive Server or ASE/CIS (as shown in the following figure):

Figure 8-2: The DB2 UDB Option for CICS and DirectConnect support components



Tracing

The DB2 UDB Option for CICS has a trace flag that will write out MFC trace points to a CICS Temp Storage Queue, the name of which consists of the two characters “CE” concatenated to the first 6 bytes of the user ID. For example, if the user name were SYBASE, the CICS Temp Storage Queue name would be CESYBASE.

For the DB2 UDB Option for CICS, you can also use the standard DirectConnect tracing support. See Figure 8-1.

Logging

You can determine the errors logged. Enter changes in SOURCE library member AMD2CAMX, as detailed in the DB2 UDB Option for CICS installation instructions. For example, to indicate that you want a message logged, set LOG=Y.

```
SYRTMAMX TYPE=ENTRY,
   MSGNR=32000,
   SQLCODE=0,
   LEVEL=11,
   STATEC=0,
   STATESC=0,
   LOG=Y,
   SQLFATAL=N
```

Warning! If a connection fails, the error log is the only method of tracking an error. To prevent the log from running out of room, initialize the log file regularly.

In a CICS environment, the DB2 UDB Option for CICS provides the *AMD2LOG* and *AMD2LOGO* files. The AMD2 transaction writes errors to *AMD2LOG*, including task ID, term ID, timestamp, and the unique message code for each error. AMD2 also returns an error message to the client.

Under CICS, if *AMD2LOG* is full or no longer accepts messages, AMD2 writes errors to the overflow file, *AMD2LOGO*. AMD2 then issues error message 33219:

```
MainframeConnect for DB2 UDB error (Write to log file
AMD2LOG failed, switching to overflow log AMD2LOGO).
```

Under CICS, you can close, archive (if desired), and purge *AMD2LOG* when it becomes full. During the time *AMD2LOG* is closed, AMD2 writes messages to *AMD2LOGO* that you can close, archive, and purge once you reopen *AMD2LOG*.

In this example, AMD2 writes an internal error 32000 to *AMD2LOG* (or *AMD2LOGO*, if *AMD2LOG* is full).

For CICS environments, you must define the error log files to CICS in an FCT entry as follows:

- *AMD2LOG*

```
DFHFCT TYPE=DATASET,
 ACCMETH=VSAM,
 DATASET=AMD2LOG,
 RECFORM=(FIXED,BLOCKED),
 DSNAME=your.syrtlog.dataset.name,
 DISP=SHR,
 SERVREQ=(ADD,BROWSE)
```

- *AMD2LOGO*

```
DFHFCT TYPE=DATASET,
 ACCMETH=VSAM,
 DATASET=AMD2LOGO,
 RECFORM=(FIXED,BLOCKED),
 DSNAME=your.syrtlogo.dataset.name,
 DISP=SHR,
 SERVREQ=(ADD,BROWSE)
```

DirectConnect communications with the DB2 UDB Option for CICS

DirectConnect depends on the communications support supplied by the platform vendor to communicate with the DB2 UDB Option for CICS. Depending on the server-based communications software installed, DirectConnect uses SNA or TCP/IP Uplink support.

SNA

SNA software uses the SNA trace file to record SDLC/SNA traffic between the server and mainframe. The vendor trace utility extracts this file.

For Windows NT platforms, the error log file records errors that SNA Server for Windows NT detects.

TCP/IP uplink

For CICS TCP/IP environments, third-party TCP/IP trace facilities provide a way of obtaining low-level TCP/IP level traces between the mainframe and DirectConnect. For AIX platforms, the error log file records errors that SNA Server/6000 detects. The IBM error log report utility extracts this information. Microsoft SNA Server for Windows NT also has a trace process.

Mainframe communications with DirectConnect

Mainframe-based communications support provides transport-level functionality. Depending on the mainframe communications software installed, Gateway-Library uses SNA or TCP/IP for z/OS.

SNA

For SNA environments, you can use the SNA General Trace Facility (GTF) files to trace SDLC/SNA traffic between Gateway-Library and the mainframe. The IBM TAP utility extracts this information.

TCP/IP for z/OS

For CICS TCP/IP environments, you can use the Netstat facility to check the status of TCP/IP connections, as well as to deactivate them if problems occur.

You can use the TCP/IP trace facility to trace traffic between DirectConnect and the mainframe.

Sybase TCP/IP Listener

The Sybase listener transaction provides the interface between CICS and z/OS/TCP. The listener enables CICS applications to use the following TCP/IP communication facilities:

- The IBM-supplied TCP/IP Listener transaction to initiate transactions when operating with TCP/IP devices, such as DirectConnect.
- The TCP/IP Sockets Interface provides CICS user exit traces, denoting each TCP call issued and the associated return call.

The Sybase listener TRACE parameter and the Server Option DEBUGSW parameter settings provide CICS user exit traces, denoting each TCP call issued and the associated return call.

CICS

For CICS, Gateway-Library tracing stores information on the TDS traffic between the mainframe and server in the VSAM ESDS file, SYTDLOG1. This information includes any errors detected in the traffic.

When turned on, the Server Option Trace flag enables the Server Option-TCP/IP interface to write user trace points to the CICS Message User log (CSML) as well as the CICS Aux trace facility.

Some TD calls fill up internal TDS buffers before sending them out to the network. For example, a TDSNDROW or TDSNDMSG call does not cause execution of a corresponding CICS EXEC SEND call, unless the TDS buffer becomes full.

Warning! To avoid losing records, periodically archive or delete the trace records on SYTDLOG1. Trace records append to this file until it is full, when it rejects any further records.

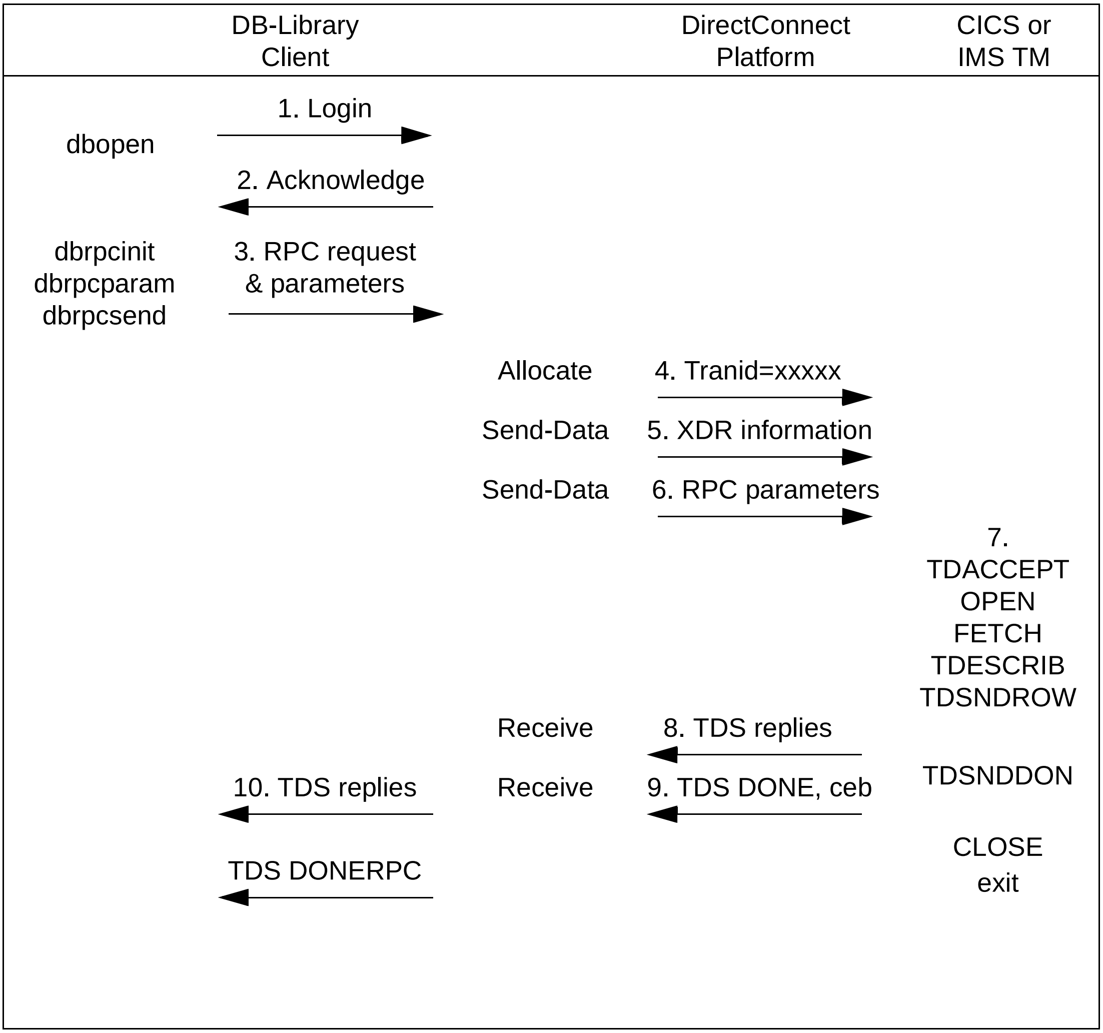
Process flow

System administrators at the mainframe, DirectConnect, and client need to coordinate troubleshooting efforts. To aid in your analysis, this section describes the process flow from the client through DirectConnect to the mainframe.

Process flow and DirectConnect requirements

DirectConnect processing can involve CT-Library, DB-Library, and ODBC. The following figure shows DirectConnect processing with DB-Library (for more information, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*):

Figure 8-3: Client-to-DirectConnect-to-mainframe processing



The following points describe the sequence shown in Figure 8-3 and highlight Enterprise Connect requirements:

Note The following points describe both RPC and dynamic SQL process flows.

1. Assuming DirectConnect is started, the client opens a LAN connection to a designated DirectConnect Service and logs in. If the following message occurs:

   ```
   Server name not found in interface file
   ```

   make sure:

   - The client *interfaces* file is correctly set up.
   - The client Sybase path variable is correctly defined.
   - The DirectConnect server is specified in the *DSQUERY* variable.

2. *For TRS only* – On receiving the client login information, DirectConnect checks security as follows:

   - If security is turned on, DirectConnect ensures the client is authorized. If the client is not authorized, the following error occurs:

     ```
     Security Violation: Login denied (no login
     entry).
     ```

   - If the client is authorized or security is turned off, DirectConnect acknowledges the login.

3. When the client application needs to invoke an RPC or language request on the mainframe, the client sends a request to DirectConnect over the LAN connection.

4. DirectConnect receives the request and searches the lookup table to find the mainframe session and the Server Option transaction ID to use. Entries for the RPC and connection must occur in the table. If security is turned on, the client must have authorization to use the RPC and connection to the mainframe. If the lookup table search and security check are successful, the line is up, and the session is active, DirectConnect allocates a conversation with the named transaction.

   If a failure occurs during this process, one of the following error messages is written to both the DirectConnect log and client.

   - If the client is not authorized or is not listed correctly, the following error message occurs:

     ```
     Security Violation: Access to RPC ‘xxxx’ denied.
     ```

   - If connections to the mainframe are unavailable, the following error message occurs:

     ```
     Request Rejected: No host connections are
     available.
     ```

- If an RPC name was entered incorrectly, or if the RPC name does not occur in the lookup table, the following message occurs:

  ```
  Request Rejected: Remote procedure 'xxxx' not
  found.
  ```

5 DirectConnect sends the client's External Data Representation (XDR) information and RPC parameters to the mainframe and waits for a reply from the transaction.

6 On the mainframe, the transaction processor initiates the named transaction, and the transaction issues the Server Option Gateway-Library calls. These calls read the client's XDR information and RPC parameters. The transaction also performs associated processing, such as the issuing of static SQL DB2 requests or reading VSAM or other database data.

7 The transaction issues Gateway-Library calls to send results back to the client. These calls perform any data conversions, generate the TDS reply datastream, and send out reply data.

8 DirectConnect receives each TDS reply packet and forwards it to the client, which continues until the Server Option transaction issues a TDSNDDON call.

  If a failure occurs during this process, an error message is written to the DirectConnect log. An "Unexpected EOF from Adaptive Server" message is written to the client. (The mainframe is acting as an Adaptive Server.)

  Gateway-Library tracing functions, if in use, also record errors in this process. See Figure 8-2.

9 When finished with the request, the transaction exits, and the conversation terminates. For a long-running transaction (also called a user-defined transaction), the transaction can remain active through multiple requests before the conversation terminates. If a long-running transaction ends before it finishes processing, you should determine whether the appropriate client support is set up:

- The client can be set up to disconnect after invoking the transaction and before the transaction ends.
- Adaptive Server may log out after sending a client request, making long-running transactions impossible.

Note AMD2 supports user-defined transactions, which are comparable to long-running transactions based on Gateway-Library support.

Process flow during attention sequences

Attention events are handled by an attention sequence. Any of the following actions can trigger an attention event:

- The Database-Library issues a dbcancel( ) command.
- A SQL user cancels processing while the server is sending results.
- An APT program or form issues a closesql command.

When an attention sequence is issued, the process flow occurs as follows:

1. Open Client issues an attention packet to DirectConnect, then discards anything else received until it receives a TDSDONE packet with the attention Ack bit on.
2. DirectConnect converts the attention packet into a SNA SIGNAL command, issuing an LU 6.2 request-to-send verb. DirectConnect then discards any results received from the mainframe until receiving a TDS DONE packet with the attention Ack bit on.
3. At the mainframe, CICS receives the SIGNAL command and informs the Server Option.
4. Gateway-Library passes back a return code, indicating TDS_CANCEL_RECEIVED, on all subsequent TDSNDROW, TDSNDMSG, and TDSETPRM calls from an application. Any data associated with TDSNDROW or TDSNDMSG calls is discarded until the application issues a TDSNDDON call.

   For details on these calls, see the Mainframe Connect Server Option *Programmer’s Reference* for COBOL or PL/1.
5. When the application issues a TDSNDDON call, the Server Option support sends a TDS DONE packet with the attention Ack bit on. This ends the attention sequence.

Procedures and common problems

This section describes procedures used and common problems encountered in troubleshooting:

- Verifying connectivity after installation
- Where to start troubleshooting

- Common problems and suggested solutions

Verifying connectivity after installation

After installing the DB2 UDB Option for CICS, make sure the AMD2 transaction processor and the client and server connections are configured correctly.

You can verify connectivity using one of the standalone utilities, snaping or cicsping, which are shipped with the Server Option to test the network connections between the workstation and mainframe. Use snaping for the Server Option for SNA and cicsping for the Server Option for TCP/IP. These utilities do not require DirectConnect to be active. For details on using them, see the Mainframe Connect DirectConnect for z/OS Option *Installation Guide*.

- If the ping utilities fail, the cause may be either a configuration error or a line outage problem. Based on the error message returned, you can narrow your search. Resolution may require coordination with the DirectConnect Administrator to make changes to the configuration.
- If the ping utilities run successfully, make sure your DirectConnect Administrator has defined all the necessary RPC and connection information to DirectConnect. Also make sure that DirectConnect is started.

Where to start troubleshooting

System administrators at the client, DirectConnect server, and mainframe must check components systematically to locate the source of a problem. Check for problems in this sequence:

1. Major outage, such as a CICS crash or power outage
2. Client application
3. Client LAN
4. Client network setup
5. DirectConnect server
6. Connection from the client application to the DirectConnect server
7. Connection from the DirectConnect server to the mainframe

8 Configuration between the transaction processor and the Server Option or the DB2 UDB Option for CICS

For any of these problems, the appropriate system administrator should:

- Record specific error message information, including:
  - Error message number
  - Associated SNA sense codes, SNA Services error codes, or TCP/IP error codes
  - The time the error occurred
  - The client or user affected
- See the appropriate documentation, as needed. For example, if you need information about the DB2 UDB Option for CICS errors, see Appendix A, “Error Messages.”
- Perform the recommended action.
- Continue the process until the problem is resolved.

Common problems and suggested solutions

Problems often can be traced to configuration errors or network, line, modem, or adapter outages.

Common problems are classified in three categories:

- Configuration errors
- Mainframe network operational failure
- Network session or line failures

Configuration errors

Configuration errors are most often the cause of communications failure. To resolve these errors, you need the following information, which is created when the network is installed and successfully implemented:

- CICS table definitions for TCT, PCT, PPT, FCT, and RCT
- SNA definitions for the logical unit (LU) and associated Logmode table
- SDLC or Token Ring connection charts to the mainframe

- Named Pipe and SPX charts to clients
- TCP/IP connection charts to clients
- For CICS LU 6.2, server SNA configuration
- For CICS TCP/IP, server TCP/IP configuration
- Sybase interface files for clients and DirectConnect servers
- Sybase security definitions, including client logins, connection groups, and transaction groups

Verify that this information has not changed since the error occurred. If there have been changes, determine if a recent change is contributing to or causing the problem. The following subsections explain the causes and actions for common configuration errors.

Cannot establish a session

Cause

Any of the following:

- Mismatched LU definitions between SNA and server
- Mismatched mode names
- Incorrect SNA MODETAB and APPLID macros
- Incorrect CICS TCT definitions

Action

Correct invalid names and information.

Coordinate with the DirectConnect Administrator to check connection and modename profiles using the snaping or cicsping utility shipped with the product.

Session established but the transaction does not run

Cause

Any of the following:

- CICS PCT or PPT entry error
- Host security error
- Incorrect transaction ID in the DirectConnect RPC table

Action

Verify definitions.

Coordinate with the DirectConnect Administrator for correct security and transaction ID setups.

SDLC line or token ring is not up

Cause: Address incorrectly configured with NCP (assumes correct line or modem setup)

Action: Check both ends of the SDLC station or Token Ring address configuration.

SDLC link and PU are active, but the LU is not active

Cause: Any of the following:

- SNA and DirectConnect LU definition errors
- SSCPID value in the local LU profile set incorrectly

Action: Use the SDLC trace and error log facilities to find the error.

Mainframe network operational failure

On the mainframe, there are two frequent causes of operational errors:

- The CICS or SNA operator varied the resource out of service.
- SNA placed the line, physical unit (PU), or LU into a non-operating (INOP) state because of a network outage.

In these cases, either:

- The DirectConnect Administrator receives SNA software time-out and connection failure messages in trying to start up DirectConnect,

  or

- The requesting client receives an SNA software message indicating that the system could not start the RPC.

When you are contacted about such messages, reactivate the necessary mainframe resources.

Network session or line failures

Line, adapter, or modem outages result in error messages at both the SNA console and DirectConnect. DirectConnect records the message and, if possible, sends a similar error message to any affected clients.

Preventative measures

Intermittent hardware errors and line degradation problems can disrupt processing but can be difficult to find. You should check periodically for these types of problems. For example:

- To check for hardware errors, use the associated SNA error logs and report errors to IBM Service.
- To check for line degradation, use SNA to periodically report the SDLC line statistics. Examine the statistics for a significant number of re-transmissions or idle detect time-outs. Line degradation results in random SDLC line failures or very slow response to the client, even during a moderate processing load.

APPENDIX A Error Messages

This appendix provides general information about error messages returned by the DB2 UDB Option for CICS and describes each message, providing a probable cause and suggested course of action.

This chapter covers the following topics:

Understanding the messages

This section provides the following information:

- Message sources
- How errors are logged
- Defining log files to CICS
- Interpreting the messages

Message sources

Two sources for the DB2 UDB Option for CICS messages are:

- DB2 UDB Option for CICS messages
- Messages mapped to Adaptive Server messages

DB2 UDB Option for CICS messages

The SQL transaction processing component of the DB2 UDB Option for CICS returns messages identified by “DB2 UDB Option for CICS error.”

These messages have a 5-digit message number and consist of two types:

- DB2 messages forwarded from DB2 to the client
- Internal messages generated by the SQL transaction processor AMD2

The AMD2 message table handles both types of messages. This message processing table, which consists of a translation table and a text table, is created during installation of the DB2 UDB Option for CICS. It is an editable table that can be customized to suit customer requirements. For more information about the AMD2 message table and customizing the DB2 UDB Option for CICS messages, see Chapter 5, “Customizing the DB2 UDB Option for CICS Messages.”

DB2 messages

DB2 errors generate messages that are returned to AMD2. The AMD2 message table processes the DB2 SQLCODEs and forwards the DB2 message and any additional information, such as SQL codes, to the client.

Internal messages

The AMD2 message table processes the DB2 UDB Option for CICS internal messages and sends them to the client.

Messages mapped to Adaptive Server messages

The DB2 UDB Option for CICS maps its messages to Adaptive Server messages where appropriate. The AMD2 message table, described in the previous section, performs this mapping. These messages are identified by 3- or 4-digit message numbers, instead of the 5-digit numbers used for the other messages.

How errors are logged

The AMD2 message table determines the logging of errors for the DB2 UDB Option for CICS. If the translation table LOG property is Y for a given message, that message writes to the *AMD2LOG* error log. Like other parts of the message table, the value of the LOG property can be changed to meet a customer’s error logging requirements.

AMD2LOG and *AMD2LOGO* are VSAM ESDS files installed as part of the product. Logged errors initially write to the *AMD2LOG* file. When the *AMD2LOG* file fills or ceases to accept messages for any reason, errors are written to the overflow file, *AMD2LOGO*.

When errors are written to the overflow file, you should close, archive (if desired), and purge the *AMD2LOG* file. While *AMD2LOG* is closed, errors write to *AMD2LOGO*. When *AMD2LOG* is reopened, the system programmer can close, archive, and purge *AMD2LOGO*. When *AMD2LOGO* is closed, errors automatically redirect to *AMD2LOG*.

Defining log files to CICS

Use the JCL member *CSDLOAD* to define log files to CICS. The *CSDLOAD* member can be found in the DB2 UDB Option for CICS JCL library.

Interpreting the messages

To help you understand what to do in response to a message, the messages are presented as follows:

Error *xxxxx* *Message text*

Explanation *Why the error occurred*

Action *The action Sybase recommends to correct the problem*

If the action recommended does not correct the problem, you should do one of the following:

- Run the error report facility for additional information about the error.
- Run a network trace.

Message Source *The source of the message*

In some cases, the message text uses different terms to refer to the products, as shown in the following table:

Table A-1: Abbreviations for message text

This abbreviation	Stands for
DB2 Access Module	DB2 UDB Option for CICS
GWLib	Gateway-Library
MSG	Transaction Router Service Library (formerly Mainframe Server Gateway)
AMD2	The name of the DB2 UDB Option for CICS transaction
AMD2LOG and *AMD2LOGO*	The names of the DB2 UDB Option for CICS primary and backup error log files

DB2 UDB Option for CICS error messages

The DB2 UDB Option for CICS error messages are listed in numerical order on the following pages.

Error 102

Message text

```
Incorrect syntax (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 103

Message text

```
String is too long (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 105

Message text

```
Unclosed quote (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 106

Message text

```
Too many table names (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 108

Message text

```
Order-by position is out of range of number of items in
the select-list (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 109

Message text

```
There are more columns in the INSERT statement than
values specified in the VALUES clause. The number of
values in the VALUES clause must match the number of
columns specified in the INSERT statement (DB2 %s
failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 131

Message text

```
Size exceeds maximum (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 147

Message text

```
Aggregate function may not appear in WHERE clause unless
in subquery in HAVING clause (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 164

Message text

```
GROUP BY expressions must refer to column names that
appear in the SELECT list (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 183

Message text

```
Scale out of range (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 189

Message text

```
Function has incorrect number of arguments (DB2 %s
 failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 191

Message text

```
SQL statement nested too deeply, too long or too complex
(DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 193

Message text

```
Object or column name is too long (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 206

Message text

```
Operand type clash (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 207

Message text

```
Invalid column name (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 209

Message text

```
Ambiguous column name (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 220

Message text

```
Arithmetic overflow (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 229

Message text

```
Permission denied (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 233

Message text

```
Column may not be null (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 241

Message text

```
Syntax error converting DATETIME from character string
(DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 242

Message text

```
Conversion to DATETIME resulted in a DATETIME value out
of range (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 254

Message text

```
Maximum number of columns exceeded (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 255

Message text

```
Maximum row size exceeded (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 257

Message text

```
Datatype conversion not allowed (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 264

Message text

```
Column name appears more than once (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 267

Message text

```
Object cannot be found (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 512

Message text

```
Subquery returned more than 1 value (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 536

Message text

```
Invalid length parameter passed to the substring
function (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 546

Message text

```
Foreign key constraint violation occurred (DB2 %s
failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 547

Message text

```
Dependent foreign key constraint violation in a
referential integrity constraint (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 550

Message text

```
The insert or update is not allowed because a resulting
row does not satisfy the view definition. (DB2 %s
failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 557

Message text

```
Cursor cannot be found (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 558

Message text

```
Attempt to open a cursor which is already open (DB2 %s
failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 559

Message text

```
Attempt to issue a cursor which is not open (DB2 %s
failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 560

Message text

```
Cursor in Update or Delete is not positioned on a row
(DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 593

Message text

```
The UPDATE WHERE CURRENT OF to a cursor failed because
the column being updated was not specified in the FOR
UPDATE OF list (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 1903

Message text

```
Index too large (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 1904

Message text

```
Too many columns for index (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 2714

Message text

```
Object already exists (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 4403

Message text

```
View is not updatable (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 4409

Message text

```
Columns in the query definition and the view definition
do not match (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 4607

Message text

```
Privilege may not be GRANTed or REVOKEd (DB2 %s failed:
%S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

AMD2 (module AMD2CAMX) call to DB2 that is mapped to an Adaptive Server message.

Error 30000

Message text

```
DB2 Access Module DBMS error (DB2 %s failed: %S)
```

Explanation

Request failed. The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

See your DB2 documentation.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 30001

Message text

```
DB2 Access Module DBMS warning (DB2 %s warning: %S)
```

Explanation

The cause is indicated by the DB2 message text and SQLCODE returned in the *%S* variable.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

No action required. This is only a warning. For more information, look up the message and SQLCODE in your DB2 documentation.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 30002

Message text

```
DB2 Access Module DBMS confirm (DB2 %s completed
successfully: %S)
```

Explanation

The DB2 request completed without errors.

- *%s* is replaced by a string message variable value.
- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.

Action

No action required.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 30003

Message text

```
DB2 Access Module DBMS info (DB2 completed SQLCODE = %d,
SQLSTATE = %s)
```

Explanation

Informational message providing the SQLCODE and statement.

- *%d* is the return code number.
- *%s* is replaced by a string message variable value.

Action

No action.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 32000

Message text

```
DB2 Access Module GWLib error (%s returned %d)
```

Explanation

An internal DB2 UDB Option for CICS error occurred.

- *%s* is replaced by a string message variable value.
- *%d* is the return code number.

Action

Record the return code, and contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Note More than one type of error can generate this number. See the following section for more information.

Error 32000

Message text

```
tds_ms error (tdinflog, etc)
```

Explanation

An internal DB2 UDB Option for CICS error occurred.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX)

Note More than one type of error can generate this number. See the previous section for more information.

Error 33214

Message text

```
DB2 Access Module error (Get storage failed)
```

Explanation

The AMD2 transaction was unable to get requested storage. CICS is running short on storage.

Action

Ask the CICS system programmer to increase the size of the DSA or EDSA storage pools in the CICS start-up procedure.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33215

Message text

```
DB2 Access Module error (Illogical request in I/O
routine. Request not honored because host cannot
continue.)
```

Explanation

A DB2 UDB Option for CICS error.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33216

Message text

```
DB2 Access Module error (Invalid request in I/O routine.
Internal error.)
```

Explanation

A DB2 UDB Option for CICS error.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33218

Message text

```
DB2 Access Module error (Rollback error)
```

Explanation

The DB2 UDB Option for CICS received an error while attempting to roll back a transaction.

Action

Check the CICS and DB2 error logs to determine the reason for the rollback error.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33219

Message text

```
DB2 Access Module error (Write to log file AMD2LOG
failed, switching to overflow log AMD2LOGO)
```

Explanation

The DB2 UDB Option for CICS is now writing messages to the secondary error log, *AMD2LOGO*. The DB2 UDB Option for CICS is unable to write to the primary error log, *AMD2LOG*, possibly because it is full.

Action

No immediate action is required. Before *AMD2LOGO* fills up, you need to determine why the write to *AMD2LOG* failed and correct the condition. If *AMD2LOG* is full, follow your site's procedures for storing error log information; then delete the messages from *AMD2LOG* and reopen it. After you reopen *AMD2LOG*, the DB2 UDB Option for CICS directs messages there.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33220

Message text

```
DB2 Access Module error (Write to overflow log file
AMD2LOGO failed. Error not logged.)
```

Explanation

The DB2 UDB Option for CICS is unable to write to the overflow error log, *AMD2LOGO*, possibly because it is full.

Action

Determine why the write to *AMD2LOGO* failed and correct the condition.

❖ **Purging an *AMD2LOGO* file**

1 Follow your site's procedures for storing message log information in *AMD2LOGO* and *AMD2LOG*.

2 Delete the archive or purge messages from *AMD2LOGO* and *AMD2LOG*.

3 Reopen *AMD2LOG* so that the DB2 UDB Option for CICS redirects messages to *AMD2LOG*.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33221

Message text

```
DB2 Access Module error (Datatype %d not supported)
```

Explanation

The specified datatype is not supported by the DB2 UDB Option for CICS for DB2. *%d* is replaced by a numeric message variable value.

Action

The client must use a supported datatype in the request.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33222

Message text

```
DB2 Access Module error (Illogical request in DB2
routine. Request not honored because DB2 cannot
continue.)
```

Explanation

An error in the DB2 UDB Option for CICS code.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33223

Message text

```
DB2 Access Module error (Invalid request in DB2 routine.
Internal error.)
```

Explanation

An internal error.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33224

Message text

```
DB2 Access Module error (No SQLDA. Internal error.)
```

Explanation

An internal error.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33225

Message text

```
DB2 Access Module error (DB2 SQL statement too long.)
```

Explanation

The SQL statement received from the client exceeds the maximum number of characters allowed by the DB2 UDB Option for CICS and DirectConnect.

Action

Requests sent by the client must be limited to fewer than 32,705 characters.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33226

Message text

```
DB2 Access Module error (DB2 Invalid character in SQL
statement at position %d)
```

Explanation

The request failed because it included an invalid character. *%d* is replaced by a numeric message variable value.

Action

Identify the invalid character and remove it; then resubmit the request.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33227

Message text

```
DB2 Access Module error (No SQL statement.)
```

Explanation

The DB2 UDB Option for CICS did not receive any SQL from the client.

Action

- If the client is an application, the client should resubmit the request.
- If the client is Adaptive Server, the system administrator should check the stored procedure for errors.
- If the client is an application other than Adaptive Server, check the application.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33228

Message text

```
DB2 Access Module error (Long varchar column data
truncated when truncation not allowed.)
```

Explanation

The DB2 UDB Option for CICS retrieved a LongVarChar column with data greater than 255 bytes when TRS was set to prevent truncation of LongVarChar columns.

Action

Reset the truncate option when you start MSG or avoid fetching columns that exceed 255 bytes.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33229

Message text

```
DB2 Access Module error (Stack overflow. Internal
error.)
```

Explanation

The DB2 UDB Option for CICS internal stack overflowed.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33230

Message text

```
DB2 Access Module error (Abend, code = %s)
```

Explanation

If the abend code is AEY9, the attachment facility did not start. *%s* is replaced by a string message variable value.

Action

For other abend codes, contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33231

Message text

```
DB2 Access Module error (Message table not loaded or
code not matched. Message code = %d, sql error = %S)
```

Explanation

The DB2 UDB Option for CICS message table (module AMD2CAMX) could not be loaded, or the message code generated within the DB2 UDB Option for CICS could not be matched in AMD2CAMX.

This message always returns in U.S. English.

- *%S* is replaced by the DB2 DSNTIAR message information, including the SQLCA message tokens and the SQLCODE.
- *%d* is replaced by a numeric message variable value.

Action

Be sure AMD2CAMX is correctly defined to CICS. If AMD2CAMX was customized at your site, be sure no message codes are removed; otherwise, contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33232

Message text

```
DB2 Access Module error (Default options will be used;
failure in %s, error = %d)
```

Explanation

An internal error in a call prevented the DB2 UDB Option for CICS from setting the options in the TDCUSTOM CSECT (control section, the Assembler verb starting the program) through the customization macro SYGWMCST. The default options are used. The DB2 UDB Option for CICS issues this message each time it is invoked until the error is resolved, but it still processes incoming SQL.

- *%s* is replaced by a string message variable value.
- *%d* is replaced by a numeric message variable value.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33233

Message text

```
DB2 Access Module error (Parsexit will not be invoked;
failure in %s, error = %d)
```

Explanation

An internal error in a Gateway-Library call specifically for the parse exit routine failed. The parse exit routine is not invoked for this instance of the AMD2 transaction. The DB2 UDB Option for CICS issues this message each time it is invoked until the error is resolved, but it still processes incoming SQL.

- *%s* is replaced by a string message variable value.
- *%d* is replaced by a numeric message variable value.

Action

Contact Sybase Technical Support.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33234

Message text

```
DB2 Access Module error (Transactions cannot be
labeled.)
```

Explanation

A transaction statement included a label. AMD2 does not support labels such as begin tran *mytran*, where *mytran* is a label for the transaction.

Action

If a transaction is in progress when AMD2 receives the invalid statement, the DB2 UDB Option for CICS ignores it and the transaction continues. If a transaction is not in progress, then AMD2 issues a rollback and a transaction does not begin.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33235

Message text

```
DB2 Access Module error (Transaction must be ended
before another transaction is begun.)
```

Explanation

A begin tran statement was sent to AMD2 before the current transaction was terminated by commit, end tran or rollback.

Action

This error message is returned to the client, but the transaction remains active, waiting for the next SQL text to be sent.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33236

Message text

```
DB2 Access Module error (BEGIN or END must be followed
by TRAN or TRANSACTION.)
```

Explanation

A begin or end statement was sent to AMD2 without tran or transaction.

Action

The DB2 UDB Option for CICS returns this error message to the client. If a transaction is in progress, it continues. If a transaction is not in progress, then another transaction does not begin.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33242

Message text

```
DB2 Access Module error (Request or RSP not found.)
```

Explanation

The DB2 UDB Option for CICS did not find the remote stored procedure (RSP) or host resident request specified in the use statement.

Action

Refer to the RSP.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33243

Message text

```
DB2 Access Module warning, Max result size, %s, exceeded
```

Explanation

The DirectConnect configured MaxResultSize or MaxRowsReturned has been exceeded. The remaining results have not been returned.

Action

Select a smaller result set or increase the configured value for MaxResultSize or MaxRowReturned.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33300

Message text

```
DB2 Access Module error (No dynamic entries available.)
```

Explanation

The DB2 UDB Option for CICS supports 1–50 dynamic requests. This limit was exceeded.

Action

Reduce the number of dynamic requests to 50 or less.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33301

Message text

```
DB2 Access Module error (Selects not allowed for dynamic
events.)
```

Explanation

The SQL entered for a dynamic event was a select statement, which is not allowed.

Action

To process a select statement, either execute it as a language request or open a cursor.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33302

Message text

```
DB2 Access Module error (Invalid cursor event, %d,
received.)
```

Explanation

An invalid cursor event was received from DirectConnect. The event number appears. This is an internal error.

Action

Report this information to your DirectConnect administrator.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33303

Message text

```
DB2 Access Module error (Unable to declare cursor; all
cursor ids are in use.)
```

Explanation

The maximum number of dynamic events are currently in a prepared state.

Action

Free at least one dynamic event name to prepare another dynamic event.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33304

Message text

```
DB2 Access Module error (Declare cursor was not a select
statement.)
```

Explanation

The SQL statement supplied when declaring a cursor must be a valid SQL select statement.

Action

Provide a valid select statement for the cursor request. For information, see your DB2 documentation.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33305

Message text

```
DB2 Access Module error (Invalid cursor ID of %d
supplied.)
```

Explanation

The cursor ID that the cursor request supplied does not match a cursor ID that was prepared by the DB2 UDB Option for CICS.

Action

Make sure that a cursor with this ID is prepared before attempting to use the ID.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33306

Message text

```
DB2 Access Module error (Unknown Open Server datatype
%d supplied.)
```

Explanation

An attempt to send data of a type that is not recognized by the DB2 UDB Option for CICS occurred.

Action

Change the client application to supply the correct datatype. See your DirectConnect documentation for more information.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33307

Message text

```
DB2 Access Module error (Cursor name of %s does not
match what was given in the update sql statement.)
```

Explanation

A cursor was prepared using one cursor name and the client tried to update the cursor using a different cursor name.

Action

Change the client application to specify the correct cursor name.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33308

Message text

```
DB2 Access Module error (bad parms in bulk load.)
```

Explanation

The DB2 UDB Option for CICS received bad parameters in a bulk load request.

Action

Try the transfer again.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33309

Message text

```
DB2 Access Module error (Invalid text pointer.)
```

Explanation

An invalid text pointer was supplied while using text functionality.

Action

Ensure that a LOB exists for this text pointer.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33310

Message text

```
DB2 Access Module error (Invalid license.)
```

Explanation

The license key is invalid or missing, or the license file is not usable.

Action

Ensure that the license file has been created, is correctly defined in CICS, and contains the correct license key.

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Error 33400

Message text

```
Sample Parsexit message for sample parse exit: SQL
Command was not a Select; batch aborted.
```

Explanation

The sample parse exit routine issues this message. The sample program rejects all incoming SQL except SELECT statements.

Action

For information about disabling the parse exit routine, see “Parse exit routine” For information about changing the AMD2 message table, see “Changing the AMD2 message table.”

Message source

DB2 Access Module - AMD2 (module AMD2CAMX).

Glossary

access code

A number or binary code assigned to programs, documents, or folders that allows authorized users to access them.

access service

A logical server application, used with an access service library, to which clients connect. Each DirectConnect for z/OS Option server can have multiple access services.

Access Service Library (ACSLIB)

A component of the DirectConnect for z/OS Option, a service library that provides access to non-Sybase data contained in a database management system or other type of repository. Each such repository is called a "target." Each access service library interacts with exactly one target and is named accordingly. See also **service library** and **access service**.

Adaptive Server Enterprise

The server in the Sybase Client-Server architecture. It manages multiple databases and multiple users, tracks the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory. Formerly known as SQL Server.

advanced program-to-program communication (APPC)

Hardware and software that characterize the LU 6.2 architecture and its various implementations in products. See also **logical unit 6.2 (LU 6.2)**.

AMD2

The default name of the CICS transaction in the DB2 UDB Option for CICS that allows clients to submit SQL statements to DB2 UDB. Compare with **SYRT**.

American Standard Code for Information Interchange

See **ASCII (American Standard Code for Information Interchange)**.

API

See **application program interface (API)**.

APPC

See **advanced program-to-program communication (APPC)**.

APPC/MVS

An SNA application that extends APPC support to the OS/390 operating system. The primary role of APPC/MVS is to provide full LU 6.2 capability to OS/390 applications to allow communication with other applications in a distributed SNA network.

Note SNA is no longer supported for the Client Option.

APPC communications link

Hardware and software configured to enable a remote transaction program to establish an APPC conversation with a partner transaction program in an SNA network. See also **Systems Network Architecture (SNA)**.

application program

A program that is specific to the solution of an application problem.

application program interface (API)

A functional interface, supplied by an operating system or other licensed program, that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

argument

A value supplied to a function or procedure that is required to evaluate the function.

ASCII (American Standard Code for Information Interchange)

A 7-bit standard code that permits transmittal of text, numbers, and some special characters among systems. Characters are represented by the numbers between 32 and 127, inclusive. The set includes uppercase and lowercase letters, numbers, and frequently-used special characters (such as $ & ; : # . ,).

batch

A group of records or data processing jobs brought together for processing or transmission.

bind

SQL statements from a compiled mainframe program where the access to DB2 UDB was optimized and saved as static SQL in an application plan.

boundary session control block (BSB)

A block of memory used to establish a connection between nodes in an SNA network architecture.

Note SNA is no longer supported for the Client Option.

C/370

An application programming language. Open Client Client-Library is available for C.

call

The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point.

character data

Data in the form of letters and special characters, such as punctuation marks.

character set A set of specific (usually standardized) characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.

character string A sequence of consecutive characters that are used as a value.

CICS See **Customer Information Control System (CICS)**.

CICS region The CICS area of the computer system in which an application is running.

client In client/server systems, the part of the system that sends requests to servers and processes the results of those requests. See also **client/server**. Contrast with **server**.

client application Software that is responsible for the user interface, including menus, data entry windows, and report formats or an application that sends requests to another application that acts as a server. See also **client**, **client/server**, **server**.

Client Option A Sybase product that provides capability for the mainframe to act as a client to LAN-based resources. See **client**.

client/server An architecture in which the client is an application that handles the user interface and local data manipulation functions, while the server provides data processing access and management for multiple clients. See also **client application**.

Client Services Application (CSA) A customer-written CICS program initiated on the host that uses the Sybase API to invoke Server Option for DB2 UDB as a client to Server Option or to SQL Server. See also **application program interface (API)**.

COBOL (common business-oriented language) A high-level programming language, based on English, that is used primarily for business applications.

code page An assignment of graphic characters and control function meanings to all code point.

command An order for an action to take place.

commit An instruction to a database to make permanent all changes made to one or more database files since the last commit or rollback operation, and to make the changed records available to other users. Contrast with **rollback**.

common area A control section used to reserve a main storage area that can be referred to by other modules.

common business-oriented language See **COBOL (common business-oriented language)**.

Common Programming Interface

The SAA Common Programming Interface (CPI) specifies the languages and services used to develop applications across SAA environments. The elements of the CPI specification are divided into two parts: processing logic and services.

compile

To translate all or part of a program that is expressed in a high-level language into a computer program that is expressed in an intermediate language, an assembler language, or a machine language.

connectivity

The capability to attach a variety of functional units without modifying them.

control section (CSECT)

The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

control statement

In programming languages, a statement that is used to alter the continuous sequential execution of statements; a control statement may be a conditional statement, or an imperative statement.

conversation

a) A dialog between a user and an interactive data processing system.

b) Within the context of APPC, an exchange of information or a sequence of messages sent between two transaction programs. Conversations take place between two LUs over an established session. Also, a sequence of messages sent between two applications (for instance, client application and SQL Server).

CSECT

See **control section (CSECT)**.

cursor

In SQL, a named control structure used by an application program to point to a row of data. The position of the row is within a table or view, and the cursor is used interactively to select rows from the columns.

Customer Information Control System (CICS)

An IBM-licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

data area

A storage area used by a program or device to hold information.

database

A set of related data tables and other database objects that are organized and presented to serve a specific purpose.

database management system (DBMS)

A computer-based system for defining, creating, manipulating, controlling, managing, and using databases. The software for using a database can be part of the database management system, or it can be a stand-alone database system.

data definition statement (DD statement)

A job control statement describing a data set associated with a specific job step. See also **job control language (JCL)**.

data object In a program, an element of data structure, such as a file, array, or operand, that is needed for the execution of a program and that is named or otherwise specified by the allowable character set of the language in which the program is coded.

data queue An object that communicates and stores data used by several programs in a job or between jobs.

data record A collection of items of information from the standpoint of its use in an application, as the user supplies it. The data record is stored physically separate from its associated control information in a control interval.

data set The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set name (DSN) The term or phrase used to identify a data set.

datatype In programming languages, a set of values together with a set of permitted operations.

DBCS See **double-byte character set (DBCS)**.

DB-Library A Sybase and Microsoft API that allows client applications to interact with ODS applications. See also **application program interface (API)**.

DBMS See **database management system (DBMS)**.

DB2 UDB An IBM relational database management system.

DB2 UDB Option for CICS A Sybase mainframe solution that provides dynamic access to DB2 UDB data. It replaces the OmniSQL Access Module for DB2 UDB-CICS and the functionality in the MDI Access Server. See also **Customer Information Control System (CICS)**, **DB2 UDB**, **Multiple Virtual Storage (OS/390)**.

DD statement See **data definition statement (DD statement)**.

delimiter A character that groups or separates words or values in a line of input.

direct access storage device (DASD) A device in which access time is effectively independent of the location of the data.

DirectConnect for z/OS Option
A Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management. Each DirectConnect for z/OS Option consists of a server and one or more service libraries to provide access to a specific data source. The DirectConnect for z/OS Option replaces the products “MDI Database Gateway” and “Net-Gateway.”

DirectConnect Manager
A Sybase Windows application that provides remote management capabilities for DirectConnect for z/OS Option products. These capabilities include starting, stopping, creating, and copying services.

directory
A type of file containing the names and controlling information for other files or other directories.

disk volume
A disk pack or part of a disk storage module.

double-byte character set (DBCS)
A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with **single-byte character set (SBCS)**.

driver
A system or device that enables a functional unit to operate.

dump
To record, at a particular moment, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging.

dynamic SQL
Pertaining to the preparation and processing of SQL source statements within a program while the program runs. The SQL source statements are contained in host-language variables rather than being coded directly into the application program. The SQL statement can change several times while the program runs. Contrast with **static SQL**.

EBCDIC (Extended Binary-Coded Decimal Interchange Code)
A coded character set of 256 8-bit characters.

embedded SQL (ESQL)
SQL statements that are embedded within a program and are prepared in the program preparation process before the program runs. After it is prepared, the statement itself does not change, although values of host variables specified within the statement might change.

enable In interactive communications, to load and start a subsystem, or to design a product in such a way as to facilitate the inclusion of national language functions.

end-of-file A coded character recorded on a data medium to indicate the end of the medium or end of data.

environment variable A variable that describes how an operating system runs and the devices it recognizes.

error log A data set or file in a product or system where error information is stored for later access.

ESQL See **embedded SQL (ESQL)**.

exit routine A user-written routine that receives control at predefined user exit points.

expression In programming languages, a language construct for computing a value from one or more operands; for example, literals, identifiers, array references, and function calls.

external call interface A CICS client facility that allows a program to call a CICS application as if the calling program had been linked synchronously from a previous program instead of started from a terminal.

FCT See **forms control table (FCT)**.

forms control table (FCT) An object that contains the special processing requirements for output data streams received from a host system by a remote session.

gateway Connectivity software that allows two or more computer systems with different network architectures to communicate. Contrast with **router**.

globalization The combination of internationalization and localization. See also **internationalization**, **localization**.

global variable A variable defined in one portion of a computer program and used in at least one other portion of the computer program. Contrast with **local variable**.

group ID A combination of alphanumeric characters that corresponds to a specific group name. The group ID can often be substituted in commands that take a group name as a value.

handler A routine that controls a program’s reaction to specific external events; for example, an interrupt handler.

hexadecimal A system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 16.

host The mainframe or other machine on which a database, an application, or a program resides.

IMS TM See **Information Management System Transaction Monitor (IMS TM)**.

Information Management System Transaction Monitor (IMS TM) A database/data communication (DB/DC) system that can manage complex databases and networks.

interface Hardware, software, or both, that links systems, programs, or devices.

internationalization The process of extracting locale-specific components from the source code and moving them into one or more separate modules, making the code culturally neutral so it may be localized for a specific culture. See also **globalization**. Contrast with **localization**.

invoke To start a command, procedure, or program.

JCL See **job control language (JCL)**.

job A collection of related programs, identified by appropriate job control statements.

job control language (JCL) In OS/390, a control language used to identify a job to an operating system and to describe the job’s requirements.

kanji A graphic character set consisting of symbols used in Japanese idiographic alphabets. Each character is represented by 2 bytes.

keyword In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; a keyword normally has the form of an identifier.

LAN See **local area network (LAN)**.

length The number of characters in a character string.

library a) A named area on disk that can contain programs and related information (not files). A library consists of different sections, called library members.

b) A partitioned data set containing file members for the mainframe.

library member A named collection of records or statements in a library.

line feed The movement of the print or display position to the corresponding position on the next line.

linkage
In computer security, combining data or information from one information system with data or information from another system with the intention to derive additional information; for example, the combination of computer files from two or more sources.

linkage editor
A computer program for creating load modules from one or more object modules or creating load modules by resolving cross references among the modules and, if necessary, adjusting addresses.

link-edit
To create a loadable computer program by means of a linkage editor. See also **linkage editor**.

load module
All or part of a computer program in a form suitable for loading into main storage for execution. A load module is usually the output of a linkage editor.

local area network (LAN)
A computer network located on the user's premises and covering a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary can be subject to some form of regulation.

localization
The process of preparing an extracted module for a target environment. The following items are addressed: messages are displayed and logged in the user's language; numbers, money, dates, and time are represented using the user's cultural convention; and documents are displayed in the user's language. See also **globalization**. Contrast with **internationalization**.

local variable
A variable that is defined and used only in one specified portion of a computer program. Contrast with **global variable**.

log file
The log file maintained by the Server Option Server. The server log file contains entries of events for each service managed by the Server Option Server.

logical unit (LU)
A type of network-accessible unit that enables end users to gain access to network resources and communicate with each other.

logical unit 6.2 (LU 6.2)
A type of logical unit that supports general communication between programs in a distributed processing environment. See also **advanced program-to-program communication (APPC)**.

login
The act of gaining access to a computer system by entering identification and authentication information at the workstation.

LU
See **logical unit (LU)**.

LU 6.2
See **logical unit 6.2 (LU 6.2)**.

macro An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that can also specify values for parameters in the replaced instructions.

mainframe A large computer, in particular one to which other computers can be connected so that they can share facilities the mainframe provides; for example, a System/370 computing system to which personal computers are attached so that they can upload and download programs and data.

mainframe access products Sybase products that enable client applications to communicate with mainframes in a client/server environment. See **client/server**.

main storage Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing.

map A set of values that have defined correspondence with the quantities or values of another set.

member A partition of a partitioned data set.

module A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

Multiple Virtual Storage (OS/390) An IBM operating system that runs on most mainframes. It supports 24-bit addressing up to 16 megabytes.

OS/390 See **Multiple Virtual Storage (OS/390)**.

NCP See **Network Control Program (NCP)**.

Net-Gateway An end-of-life Sybase product that provided communication between a mainframe and a LAN server. Net-Gateway was the “ancestor” of the DirectConnect for z/OS Option Transaction Router Service.

nest To incorporate one or more structures of one kind into a structure of the same kind; for example, to nest one loop (the nested or inner loop) within another loop (the nesting or outer loop); to nest one subroutine within another subroutine.

network A configuration of data processing devices and software connected for information exchange.

Network Control Program (NCP) An IBM licensed program that provides communication controller support for single domain, multiple-domain, and interConnected network capability.

null A pointer that does not point to a data object.

object A passive entity that contains or receives information but cannot change the information it contains. In Server Option, objects include rows, tables, databases, stored procedures, triggers, defaults, and views.

object code Output from a compiler or assembler that is also executable machine code or is suitable for processing to produce executable machine code. Contrast with **source code**.

ODBC See **Open Database Connectivity (ODBC)**.

OmniConnect Translates Sybase SQL syntax into statements that DB2 UDB can process.

Open Client A Sybase product that provides customer applications, third-party products, and other Sybase products with the interfaces required to communicate with Open Client and Open Server applications.

Open Client application An application written using Open Client libraries.

Open Database Connectivity (ODBC) A Microsoft API that allows access to both relational and non-relational databases. ODBC allows client application developers to produce vendor-neutral Windows applications that can access data sources without including code for a specific database. See also **application program interface (API)**.

Open Server A Sybase product that provides the tools and interfaces required to create a custom server. For example, clients can route requests to the DirectConnect for z/OS Option, which is an Open Server application that they configured to meet specific needs, such as the preprocessing of SQL statements or decision making about routing RPCs to Transaction Router Service for DB2 UDB or to other servers. See the Open Server documentation for information about this product.

OS PL/1 Version II An application programming language. Open Client Client-Library and Open Server Gateway-Library are both available for PL/1.

overwrite To write into an area of storage, thereby destroying the data previously stored in the same area.

parameter A variable that is given a constant value for a specified application and that can denote the application. Contrast with **property**.

parse In systems with time sharing, to analyze the operands entered with a command and create a parameter list for the command processor from the information.

PARTNER table A CICS table through which the CPI-C maps

pipe
To direct data so that the output from one process becomes the input to another process. The standard output of one command can be connected to the standard input of another with the pipe operator (|). Two commands connected in this way constitute a pipeline.

PL/1
See **Programming Language/I (PL/1)**.

pointer
A data element that indicates the location of another data element.

precompile
To process programs containing SQL statements before they are compiled. SQL statements are replaced with statements that will be recognized by the host language compiler. The output from this precompile includes source code that can be submitted to the compiler and used in the bind process.

Programming Language/I (PL/1)
A programming language designed for use in a wide range of commercial and scientific computer applications.

property
A setting for a server or service that defines the characteristics of the service, such as how events are logged or how datatypes are converted. Contrast with **parameter**.

protocol
A set of rules that governs the behavior of computers communicating on a network.

pseudocode
A set of instructions that is logically structured but does not follow the syntax of any particular programming language.

RDBMS
See **relational database management system (RDBMS)**.

relational database
A database in which data is viewed as being stored in tables consisting of columns (data items) and rows (units of information). Data from different tables can be combined to form new data relationships.

relational database management system (RDBMS)
An application that controls relational databases. See also **relational database**. Contrast with **database management system (DBMS)**.

remote procedure call (RPC)
A stored procedure executed on a different Server Option server from the one onto which a user is logged.

remote stored procedure (RSP)
A customer-written CICS program that resides on the mainframe and communicates with Server Option for CICS. See also **Customer Information Control System (CICS)**. Contrast with **Client Services Application (CSA)**.

resource table
A main storage table that associates each resource identifier with an external logical unit (LU) or application program.

return code A value returned to a program to indicate the results of an operation requested by that program.

rollback An instruction to a database to back out of the changes requested in a unit of work. Contrast with **commit**.

router An attaching device that connects two LAN segments, which use similar or different architectures, at the OSI reference model network layer. Contrast with **gateway**.

routine A program, or part of a program, that can have general or frequent use.

RPC See **remote procedure call (RPC)**.

RSP See **remote stored procedure (RSP)**.

SAA See **System Application Architecture**.

SBCS See **single-byte character set (SBCS)**.

server A functional unit that provides shared services to workstations over a network. Contrast with **client**. See **client/server**.

Server Option A Sybase product that provides capability for programmatic access to mainframe data.

service A functionality available to DirectConnect for z/OS Option applications. It is the pairing of a service library and a set of specific configuration properties.

service library A set of configuration properties that determine service functionality. Examples of service libraries include access service libraries, transfer service libraries, administrative service libraries, and transaction router service libraries. See also **Access Service Library (ACSLIB)**.

shell A command interpreter that acts as an interface between the user and the operating system. A shell can contain another shell nested inside it; the outer shell is the parent shell, and the inner shell is the child.

single-byte character set (SBCS) A character set in which each character is represented by a 1-byte code. Contrast with **double-byte character set (DBCS)**.

SNA See **Systems Network Architecture (SNA)**.

socket A unique host identifier created by the concatenation of a port identifier with a TCP/IP address.

source code The input to a compiler or assembler, written in a source language. Contrast with **object code**.

source language A language from which statements are translated.

SPAREA (Stored Procedure Communication Area) An area in which a CSA exchanges information with the Client Option.

SQL See **structured query language (SQL)**.

SQLDA (SQL descriptor area) A set of variables used in the processing of certain SQL statements. The SQLDA is intended for dynamic SQL programs.

SQL descriptor area See **SQLDA (SQL descriptor area)**.

SQL Server See **Adaptive Server Enterprise**.

staging The movement of data from an off-line or low-priority device back to an online or higher-priority device, usually on demand of the system or on request of a user.

stand alone An operation that is independent of any other device, program, or system.

standard input (STD input) The primary source of data entered into a command. Standard input comes from the keyboard unless redirection or piping is used, in which case standard input can be from a file or the output from another command.

statement A basic unit of SQL, which is a single SQL operation, such as select, update, or delete.

static SQL SQL statements that are embedded within a program and are prepared during the program preparation process before the program runs. After being prepared, the statement itself does not change, although values of host variables specified by the statement can change. Contrast with **dynamic SQL**.

STD input See standard input.

Stored Procedure Communication Area See **SPAREA (Stored Procedure Communication Area)**.

string In programming languages, the form of data used for storing and manipulating text. For example, in PL/1, a string is a sequence of characters or bits that is treated as a single data item; and in SQL, a string is a character string.

structured field A mechanism that permits variable-length data or non-3270 data to be encoded for transmission in the 3270 data stream.

structured query language (SQL) An IBM industry-standard language for processing data in a relational database.

stub A program module that transfers remote procedure calls and responses between a client and a server. See **client**, **server**.

SYRT SYRT is the default name of the CICS transaction in Omni SQL Access Module for DB2 UDB that allows clients to submit SQL statements to DB2 UDB. It has been replaced by **AMD2** (the DB2 UDB Option for CICS).

System Application Architecture SAA is an architecture composed of a set of selected software interfaces, conventions, and protocols designed to provide a framework for developing distributed applications. The key benefits of SAA are: portability, consistency, and connectivity. The components of SAA are specifications for the key application interfaces points: common user access, common communication support, and common programming interface.

Systems Administrator A user authorized to handle Server Option system administration, including creating user accounts, assigning permissions, and creating new databases.

Systems Network Architecture (SNA) An IBM proprietary plan for the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling network configuration and operation. See also **advanced program-to-program communication (APPC)**.

Note SNA is no longer supported for the Client Option.

Systems Programmer A programmer who plans, generates, maintains, extends, and controls the use of an operating system with the aim of improving overall productivity of an installation.

table An array of data or a named data object that contains a specific number of unordered rows. Each item in a row can be unambiguously identified by means of one or more arguments.

tabular data stream (TDS) The proprietary Sybase protocol that defines the format of data transmitted between client and server programs in an efficient, self-describing manner.

temporary storage In computer programming, storage locations reserved for intermediate results.

transaction An exchange between a program on a local system and a program on a remote system that accomplishes a particular action or result.

transfer A DirectConnect for z/OS Option feature that allows users to move data or copies of data from one database to another.

transient A program or subroutine that does not reside in main storage or in a temporary storage area for such a program.

Transaction Router Service (TRS)
A DirectConnect for z/OS Option product used when the mainframe acts as a transaction server to route requests from remote clients to a mainframe transaction and return results to the clients. See also **DirectConnect for z/OS Option**.

Transmission Control Protocol/Internet Protocol (TCP/IP)
A set of communication protocols that supports peer-to-peer connectivity functions for both local and wide area networks.

troubleshoot
To detect, locate, and eliminate errors in computer programs or faults in hardware.

TRS
See **Transaction Router Service (TRS)**.

variable
An entity that is assigned a value.

VS COBOL II
An application programming language. Open Client Client-Library and Open Server Gateway-Library are both available for COBOL.

workstation
A terminal, microcomputer, or personal computer usually one that is connected to a mainframe or to a network, at which a user can perform tasks.

z/OS
An IBM operating system that runs on most mainframes. It supports 24-bit addressing up to 16 megabytes. See **Multiple Virtual Storage (OS/390)**.

Index