SYBASE®

Installation and Administration Guide

Mainframe Connect Server Option

12.6

[IBM IMS and MVS]

DOCUMENT ID: DC34368-01-1260-01

LAST REVISED: May 2005

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

Contents

About This Book

The Mainframe Connect Server Option for IMS and MVS *Installation and Administration Guide* describes how to install and configure the Server Option for IMS and MVS. It also addresses system administration.

This preface includes the following topics:

Note If you want to go directly to the installation instructions, skip to Chapter 2, "Planning Your Installation."

Audience

The guidelines and instructions in this book are intended for those who install, configure, and maintain Sybase® mainframe components on an IBM z/Series mainframe computer. This book refers to anyone performing these tasks as the Server Option administrator.

To use this book, you should have a working knowledge of system administration for your environment.

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

Old product names	New product name
• 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	Mainframe Connect DirectConnect for z/OS Option

The new product names are used throughout this book.

How to use this book

The following table shows how this book is organized.

To		See
Understand	The Server Option	Chapter 1, “Understanding the Server Option”
Plan	The Server Option installation	Chapter 2, “Planning Your Installation”
Install	The Server Option	Chapter 3, “Installation and Configuration”
Understand	Server Option security	Chapter 4, “Security”
Set up	Tracing and accounting	Chapter 5, “Tracing and Accounting”
Reference	Translation tables	Appendix B, “Translation Tables”
Customize	The Server Option	Appendix A, “Customization Options”
Troubleshoot	Problems with client access to data	Appendix C, “Troubleshooting”

Related documents

To install and use the Server Option, you may need to refer to the following documentation:

- Mainframe Connect Server Option *Programmer’s Reference for PL/1*
- Mainframe Connect Server Option *Programmer’s Reference for COBOL*

- Mainframe Connect Server Option *Programmer's Reference for Remote Stored Procedures*
- Mainframe Connect Client Option *Programmer's Reference for PL/1*
- Mainframe Connect Client Option *Programmer's Reference for COBOL*
- Mainframe Connect Client Option *Programmer's Reference for C*
- Mainframe Connect Client Option *Programmer's Reference for Client Services Applications*
- Mainframe Connect Client Option and Server Option *Messages and Codes*
- 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*

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

The Server Option uses 8-character function names; other versions of Server-Library use longer names. This book uses the long version of Server-Library names with this exception: the 8-character version is used in syntax statements. For example, in a syntax statement, "CTBCMDPROPS" is written "CTBCMDPR." You can use either version in your code.

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	
( )	When you see parentheses, include them 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 one or more of the enclosed options, or none. 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.

This book uses the following style conventions:

This type of information	Looks like this
Gateway-Library function names	TDINIT, TDRESULT
Client-Library™ function names	CTBINIT, CTBRESULTS
Other executables (DB-Library™ routines, SQL commands) in text	the dbrpcparam routine, a select statement

This type of information	**Looks like this**
Directory names, path names, and file names	*/usr/bin directory, interfaces* file
Variables	*n* bytes
Datatypes	datetime, float
Sample code	01 BUFFER PIC S9(9) COMP SYNC
User input	*01 BUFFER PIC X(n)*
Client-Library and Gateway-Library function argument names	*BUFFER*, *RETCODE*
Names of objects stored on the mainframe	SYCTSAA5
Symbolic values used with function arguments, properties, and structure fields	CS-UNUSED, FMT-NAME, CS-SV-FATAL
Client-Library property names	CS-PASSWORD, CS-USERNAME
Client-Library and Gateway-Library datatypes	CS-CHAR, TDSCHAR

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 Server Option

This chapter includes the following topics:

What is the Server Option?

The Server Option is an application programming environment (API) enabling the creation of mainframe applications for use by Sybase client applications. Server Option applications can retrieve and update data stored in mainframe resources like the following:

- DB2 UDB and other relational database management systems (RDBMSs)
- DL/1 databases
- Transient Storage (TS) queues
- Transient Data (TD) queues
- VSAM files

The Server Option is available for CICS, IMS TM, and MVS.

Note This book explains how the Server Option functions in an IMS or MVS environment. For information on how the Server Option functions in the CICS environment, see the Mainframe Connect Server Option for CICS *Installation and Administration Guide*.

Architecture

The Server Option runs on an IBM z/Series or plug-compatible mainframe computer. The Server Option uses LU 6.2, CPI-C, or IBM TCP/IP communications with a host transaction processor, such as MVS, as a communications front end.

Note The Server Option for IMS and MVS functions only in a three-tier (gateway-enabled) environment.

When installing and using the Server Option, follow the instructions in this book for your environment.

Functionality

Server Option applications can receive requests from LAN clients and Client Option applications in a three-tier environment using the DirectConnect for z/OS Option DB2 access service or Transaction Router Service (TRS).

Requests in a three-tier environment

In the Server Option network configuration using a three-tier (gateway-enabled) SNA environment, the DirectConnect for z/OS Option accepts requests from LAN-based clients and routes them to the appropriate server.

Server Option applications receive requests from LAN clients through either of the following DirectConnect for z/OS Option components:

- DirectConnect for z/OS Option access service
- TRS

DirectConnect for z/OS Option access service

An access service is a logical server application, used with an access service library, that enables a LAN client to communicate with Server Option applications. Each DirectConnect for z/OS Option server can have multiple access services.

For more information about access services, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for DB2 Access Services*.

TRS

TRS enables Sybase clients running on workstations and sharing a local area network (LAN) to access mainframe data and applications. The TRS listener waits for and accepts client requests and routes them to the mainframe, using transaction and connection information the DirectConnect administrator provides during configuration.

TRS treats all client requests like remote procedure calls (RPCs). TRS maps each request to a specific mainframe transaction. On receiving a client request, TRS invokes the corresponding mainframe transaction. The transaction processor runs the transaction and returns results to TRS, which forwards the results to the requesting client.

For details, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

Configuration in a three-tier architecture

The mainframe and TRS configuration parameters must be coordinated to permit communication with one another. When configuring a mainframe region to communicate with TRS, coordinate the following mainframe configuration values with TRS:

- For LU 6.2:
  - MVS connection and session definitions
  - Virtual Telecommunications Access Method (VTAM)
  - Network Control Program (NCP)
  - SNA, using your TRS platform SNA support program
- For TCP/IP:
  - TCP/IP for z/OS port definitions
  - Sybase listener configuration values

Note The configuration values are provided in the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

A Sybase network can also connect multiple clients and multiple servers, running in one or more MVS regions on one or more mainframes. MVS accepts simultaneous client requests through TRS and assigns them to host resources based on availability. MVS also provides transaction management services, including the handling of synchronization points, rollbacks, commits, and recovery.

Note Client applications can access the Server Option with TCP/IP in a three-tier environment by using IBM IMS Connect.

Compatibility with other products

For full functionality with the current release, use these Sybase components, as available at your site:

Table 1-1: Sybase product release compatibility

Component	Release level
Client Option for IMS and MVS	12.5, 12.6
DirectConnect Transaction Router Service	12.5, 12.6

CHAPTER 2 Planning Your Installation

This chapter explains general criteria for choosing a driver, provides an overview of the installation process, and describes pre-installation tasks. It includes the following topics:

Choosing a network driver

The Server Option provides added flexibility and easy installation for sites configured to run SNA, CPI-C, and TCP/IP by supporting the concurrent use of multiple network drivers. Programs can invoke network drivers from the same Server Option and Client Option common code base, and the appropriate network driver loads dynamically during program execution.

General criteria for choosing a driver

The choice of a network driver depends on your network type and operating environment.

IMS and MVS environment

The following drivers are supported in the IMS or MVS environment:

- CPI-C for an SNA network
- LU 6.2 for an SNA network

The following table indicates which drivers can be used by the Server Option for IMS and MVS in a three-tier environment.

Driver	IMS	MVS
CPICIMS	X	
CPICMVS		X
LU62IMS	X[1]	

[1] Client applications may access the Server Option for IMS if IBM IMS Connect listens at the mainframe end. In this case, the implicit mode driver LU62IMS is used.

Note The Server Option supports TCP/IP access to IMS using the IBM IMS Connect TCP/IP listener and the Mainframe Connect DirectConnect for z/OS Option (version 12.5.3 and later). For information on enabling TCP/IP access to IMS, see "Post-installation steps" in Chapter 3, "Installation and Configuration."

Planning the installation

This section includes the following topics:

- Installation media
- Pre-installation tasks

Installation media

The Server Option is 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 IMS and MVS *Release Bulletin*.

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 the platforms, components, and distributed software
2. Verify the space requirements
3. Determine JCL and system information
4. Determine CICS and DB2 UDB information
5. Determine compiler information
6. Determine Server Option information
7. Determine ftp information
8. Plan the security requirements
9. Identify the change control requirements
10. Back up the release libraries (upgrades only)
11. Determine the library names
12. Verify the connectivity

Task list

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

1. Verify the platforms, components, and distributed software

See the Mainframe Connect Server Option for IMS and MVS *Release Bulletin*.

2. Verify the space requirements

Verify the space required install the Server Option. The total space required for the Server Option is approximately 15.5MB.

Note You can duplicate various libraries to support the configuration of your environment. However, duplication requires additional space.

3. Determine JCL and system information

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

- JCL jobcard values. These are used in the final installation jobs run in TSO.
- High-level qualifier. This is used as a prefix for data sets generated during installation.
- Volume serial number. This indicates where generated data sets are cataloged.
- Unit parameter value. This 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.

4. Determine CICS and DB2 UDB information

Determine the following information if you intend to install a component that uses CICS or DB2 UDB:

- High-level qualifier for CICS system data sets.
- RDO data set name. This is the name of the CICS RDO (DFHCSD) containing the application resource definitions used by your CICS region.
- RDO group list. This 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. This is the high-level qualifier used for DB2 system data sets.
- DB2 exit data set name (DSN).
- DB2 DSN.

5. Determine compiler information

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

- LE370 high-level qualifier, 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.
- PLI compiler name, which is the module used to execute PLI in your environment.
- PLI compiler loadlib, which is the system loadlib where your PLI compiler module resides.
- C compiler data sets high-level qualifier. This is the high-level qualifier used for C.
- TCP/IP data sets high-level qualifier.

6. Determine Server Option information

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

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

7. 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. This 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. This indicates where ftp log information is to be written.

8. Plan the security requirements

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

9. 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, OS/390, security, change control, and scheduling
- A schedule, including cut-off dates for specific tasks

10. Back up the release libraries (upgrades only)

If you are upgrading an existing release of the Server Option, Sybase strongly recommends that you back up the entire set of release libraries before beginning this installation.

11. 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 releases 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 continue to use the old Sybase library names, delete all members before installing the new ones with the new version.

12. Verify the connectivity

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

CHAPTER 3 Installation and Configuration

This chapter describes the following topics.

Before you begin

Be sure you completed the tasks in Chapter 2, “Planning Your Installation.”

Installation and configuration

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 IMS and MVS. 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 for the Server Option for IMS. If you are installing the Server Option for MVS, skip to the next procedure.

❖ Completing the installation for IMS

1. Locate the installation JCL for the Server Option for IMS in *hlq.OSC126.IMS.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.
   - *IxHOST*: This job assembles and links the Server Option customization module, character sets, licensing, and remote host definitions. You may rerun this job at any time to change configuration and character sets or to add, remove, or modify remote host definitions.
   - *IxPSB*: This job generates PSBs.
   - *IxCAPPC*: This job defines APPC VSAM datasets.
   - *IxUAPPC*: This job inserts entries into the APPC VSAM datasets.
   - *IxDELETE*: This optional job deletes the data sets in the TSO XMIT format used for the installation.

3 Run the following jobs if you wish to compile and link edit the sample applications provided with the Server Option for IMS:

- *P1RDO*: This job creates sample CICS LU 6.2 RDO entries to be used with IMS.
- *SAPBIND*: This job contains the DB2 binds for sample programs that use DB2.
- *SAPBMPC*: This job contains the BMP region JCL for the SYICSAV2 sample application.
- *SAPBMPP*: This job contains the BMP region JCL for the SYIPSAV1 sample application.
- *SAPCOB1*: This job compiles and links COBOL samples that do not use DB2.
- *SAPCOB2*: This job compiles and links COBOL samples that use DB2.
- *SAPDB*: This job generates IMS data for sample applications.
- *SAPIMSA*: This job compiles and links the IMSASYCH sample application.
- *SAPPLI1*: This job compiles and links PL/1 samples that do not use DB2.
- *SAPPLI2*: This job compiles and links PL/1 samples that use DB2.
- *SAPTPPRF*: This job contains TP profiles for implicit and explicit sample applications.

❖ Completing the installation for MVS

1 Locate the installation JCL for the Server Option for MVS in *hlq.OSC126.MVS.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.

- *IxHOST*: This job assembles and links the Server Option customization module, character sets, licensing, and remote host definitions. You may rerun this job at any time to change configuration and character sets or to add, remove, or modify remote host definitions.
- *IxCAPPC*: This job defines APPC VSAM datasets.
- *IxDELETE*: This optional job deletes the data sets in the TSO XMIT format used for the installation.

3 Run the following jobs if you wish to compile and link edit the sample applications provided with the Server Option for MVS:

- *P1RDO*: This job creates sample CICS LU 6.2 RDO entries to be used with MVS.
- *SAPAPPC*: This job contains APPC TP profiles for the sample applications.
- *SAPBIND*: This job contains the DB2 binds for the sample applications that use DB2.
- *SAPCOB1*: This job compiles and links non-VSAM COBOL sample applications.
- *SAPCOB2*: This job compiles and links COBOL sample applications that use DB2.
- *SAPPLI1*: This job compiles and links non-VSAM PL/1 sample applications.
- *SAPPLI1*: This job compiles and links PL/1 sample applications that use DB2.

Post-installation steps

The Server Option supports TCP/IP access to IMS using the IBM IMS Connect TCP/IP listener and the Mainframe Connect DirectConnect for z/OS Option (version 12.5.3 and later). To enable TCP/IP access to IMS, use the following procedure.

❖ Setting up TCP access to IMS

1 Copy the supplied IMS Connect exit, named SYBTRSEX, from *hlq.OSC126.IMS.LOADLIB* to an APF-authorized library.

2 Add the name of the APF-authorized library from step 1 to the STEPLIB of the IMS Connect startup procedure. For example,

```
//IMS7HW01 PROC
//*
//STEP1 EXEC PGM=HWSHWS00,REGION=7M,TIME=1440,
// PARM='BPECFG=BPECFG01,HWSCFG=HWSCFG01'
//STEPLIB DD DSN=IMS710.HWS110.SHWSRESL,DISP=SHR
// DD DSN=IMS710B.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS710B.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//HWSRCORD DD DSN=IMS710B.HWSRCDR,DISP=SHR
```

Here, APF-authorized library is named *IMS710B.SDFSRESL*.

3 Add SYBTRSEX to the IMS configuration member specified in the startup procedure. For example,

```
HWSCFG01

HWS (ID=IMS71B01,RACF=Y,XIBAREA=20)
TCPIP (HOSTNAME=TCPIP,ECB=Y,RACFID=DEFAULT,PORTID=(8880),MAXSOC=500,
TIMEOUT=30000,EXIT=(SYBTRSEX))
DATASTORE (ID=IMSB,GROUP=IMSXCF,MEMBER=IMSHWS1,TMEMBER=IMS7B)
```

Here, SYBTRSEX is added to the *HWSCFG01* member of *IMS710B.PROCLIB.*

4 Have your DirectConnect for z/OS TRS administrator add an IMS region with the same listener port as the PORTID specified in the IMS Connect configuration member. For example,

```
1> exec sgw_addregion IMSB,ophelia,'8880',IMS
2> go
```

Note The DirectConnect for z/OS Option (version 12.5.3 and later) and IBM IMS Connect support only implicit IMS transactions for the Server Option. Explicit IMS transactions are not supported by the IBM IMS Connect TCP/IP listener for the Server Option.

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

Security

This chapter provides guidelines for setting up the Server Option security for the IMS TM environment. The information in this chapter applies to the set of DB2 UDB Option for CICS components that make up your LAN-to-mainframe configuration.

This chapter includes the following topics:

- Sybase components and security
- Security for APPC/IMS
- Implementing conversational security
- External security systems

Sybase components and security

Sybase components can provide their own security at these levels, which are explained in the following subsections:

- Client workstation
- Adaptive Server Enterprise
- TRS
- IMS TM
- Vendor SNA support software

Note Coordinate efforts to set up and maintain security between these components and communicate changes when they occur.

Client workstation

Most workstations have a secure login that verifies the identity and authorization of the user by requiring a unique user ID and password. Client user ID, password, and profile information can be passed to Adaptive Server Enterprise and to the DirectConnect for z/OS Option.

Adaptive Server Enterprise

Adaptive Server Enterprise (formerly called SQL Server®) can grant or deny a user the permission to call a particular remote procedure. Requests routed through Adaptive Server Enterprise undergo security checks. The TRS administrator can apply this security to all requests by specifying the -D (indirect access) parameter when starting TRS. This parameter requires all client requests to go through the Adaptive Server Enterprise. For details, refer to the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

Note Routing transactions in a server to server mode (such as through an ASE to the mainframe) *requires* a three-tier processing environment.

TRS

This section addresses:

- Defining security
- Overriding security
- Conversation-level security

Defining security

Under TRS security, every client login must be defined to TRS. For each client, this login definition:

- Specifies the client login ID and password
- Optionally specifies a host login ID and password
- Includes lists of the connections and host transactions available to clients using the login

By working with mainframe systems programmers and security administrators, and by carefully defining user IDs, host IDs, transactions, and connections, the TRS administrator can restrict client access to particular host resources.

When the TRS administrator defines a remote procedure call (RPC) to TRS, these security options are available:

- None
- User ID
- Both (user ID and password)

Each option represents the type of login information that is passed to the mainframe when a client calls the RPC. The TRS administrator sets security parameters for a transaction when defining the associated RPC to TRS.

When security is enforced at TRS, the mainframe verifies that the caller is authorized to access the requested transaction. If proper authorization exists, the transaction executes; otherwise, the mainframe returns an error message to the user.

Overriding security

The TRS administrator can override TRS security by setting the Security configuration property to No in the TRS configuration file. This allows you to map users to transaction groups that allow specific RPCs. For more information about what the security parameter does, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

Conversation-level security

You can set up conversation-level security, a process by which TRS passes client login information to the mainframe when it allocates a conversation. Under conversation-level security, the following can be passed to the host:

- A pre-defined host ID and password, which can be set up in the login definition
- A separate ID and password attached to the transaction group of the client

IMS TM

For detailed information, refer to the appropriate IBM documentation.

You can use RACF to check whether the inbound request is authorized to access the local LU. However, if you are using the Adapter, it sends only the password, not the user ID. You can use the Gateway-Library function TDGETUSR to retrieve the user ID from the client login packet. For details on TDGETUSR, see the appropriate Mainframe Connect Server Option *Programmer's Reference*. PL/1 and COBOL versions of this guide are available.

Vendor SNA support software

The SNA support software of the vendor may send login information to the host in FMH-5 fields with client requests. This allows you to use external security products that require client login information.

Security for APPC/IMS

APPC/MVS provides APPC/IMS TM security. The security mechanisms fall into two categories:

- LU security – to protect logical units
- Conversational security – to protect transaction programs

LU security

LU security includes:

- Session level security, which uses the RACF resource class APPCLU to set up session keys for LUs. If two LUs try to initiate a session and the session keys do not match, SNA and RACF does not allow the session to occur.
- Control of user access to LUs through the RACF class APPL.
- Control of user access from an LU through the RACF class APPCPORT.

For further details, refer to the appropriate IBM documentation.

Conversational security

Conversational security uses the RACF resource class APPCTP to determine the user IDs that can access APPC/MVS TP profiles and their associated transaction program names (TPNs).

Conversational security depends on the level of security defined in the SNA APPL statement for the APPC LU. Based on the level of security you specify, MVS checks every transaction initiated from an LU 6.2 workstation. The following table shows the levels of security that you can define on the SECACPT parameter in the SNA APPL statement.

IMS TM adapter

The Adapter sends only the password, not the user ID. To do a security check, you can use the Gateway-Library function TDGETUSR to retrieve the user ID from the client login packet. For details on TDGETUSR, see the appropriate Mainframe Connect Server Option *Programmer's Reference*. PL/1 and COBOL versions of this guide are available.

Implementing conversational security

Conversational security (CONV) requires changes to APPC/MVS, RACF, and SNA. This section addresses Server Option-specific requirements. For details on IMS TM and APPC/MVS security, see the appropriate IBM documentation.

APPC/MVS definitions

Define a token for the TP profile dataset that corresponds to the APPC/IMS TM LU. Use the DBMODIFY command of ATBSDFMU utility as follows:

```
//STEP1   EXEC PGM=ATBSDFMU
 //STEPLIBDD DSN=IMS41.RESLIB,DISP=SHR
 //SYSPRINTDD SYSOUT=A
 //SYSSDLIBDD DSN=SYS1.APPCTP,DISP=SHR
 //SYSSDOUTDD SYSOUT=*
 //SYSIN  DD  *
   DBMODIFY
                 DBTOKEN(SYBAPPC)
 /*
 //
```

RACF definitions

To set up RACF for the Server Option, do the following:

1. Define a resource profile DBTOKEN.X.TPNAME in the APPCTP class for each TPNAME defined in the TP profile dataset.
2. Add the user IDs/groups to the access list.
3. Activate and refresh the APPCTP class.

SNA definitions

Define the SECACPT=CONV parameter on the APPL definition for the APPC/IMS LU.

In addition, to allow an LU to support conversation level security, you must set the PSERVIC parameter of a logmode entry. Assign each LU a logmode corresponding to the desired level of security.

The 10th byte of PSERVIC determines security as follows:

- x’00’ – LOCAL
- x’12’ – IDENTIFY
- x’10’ – VERIFY

Example: conversation level security

Setting up a successful security system for use with MAP in an IMS TM LU 6.2 environment requires careful synchronization between SNA and IMS TM. For conversation level security, at the mainframe follow these steps:

- In the SNA LOGMODE entry, define security. VERIFY requires a user ID and password. This example highlights security defined as VERIFY:

```
SYBTABVMODETAB
   EJECT
   TITLE 'M6P1024V'
 *--------------------------------------------------------------*
 * LU6.2,SINGLE SESSIONS, RU_SIZE(1024),
 * SYNCH_LEVEL(CONFIRM), SECURITY(VERIFY)
 *--------------------------------------------------------------*
 M6P1024VMODEENT LOGMODE=M6S1024V,FMPROF=X'13',TSPROF=X'07',
   PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'78A5',
    RUSIZES=X'8787',TYPE=X'00',
    PSNDPAC=X'05',SRCVPAC=X'05',SSNDPAC=X'05',
    PSERVIC=X'060200000000000000102F00'
*MODEEND
```

Note The 10th byte of the PSERVIC parameter must be set to x’10’ for VERIFY security.

- In your network definition to SNA, specify the mode you defined in the LogMode entry. You can apply the LogMode entry to a specific LU statement, or apply it globally through the PU statement, as in this example:

```
SYBPU1PUCUADDR=041,DLOGMOD=M6P1024V,MAXBFRU=11,SSCPFM=FSS, +
       USSTAB=ISTINCDT,DELAY=0,SECNET=YES,ISTATUS=ACTIVE
*
 SYBLU02LU LOCADDR=2
```

External security systems

Most mainframe-based external security systems, such as RACF, are based on user login information. The system employs user ID and password information, restricting transaction access to authorized users.

CHAPTER 5 Tracing and Accounting

The Server Option provides functions for tracing and accounting. This chapter describes how these functions work under IMS TM and MVS and includes the layout of the logs.

This chapter contains the following topics:

- Tracing
- Accounting

Tracing

The Server Option provides tracing functions for tracing program activity, either for all transactions (global tracing) or for individual transactions (specific tracing). The Server Option writes header and data information to the error log under IMS TM in the IMS TM log API traces, which trace calls from the client application to the Server Option.

This section contains the following subsections:

- The Server Option trace functions
- Trace log
- Using the tracing facility

The Server Option trace functions

The Server Option trace functions allow you to do three types of tracing:

- API tracing, which traces Server Option calls
- TDS header tracing

- TDS data tracing

Note Tabular Data Stream (TDS)™ is the proprietary Sybase protocol that defines the format of data transmitted between client and server programs in an efficient, self-describing manner.

You can enable and disable any kind of transaction tracing globally or specifically. Use these Server Option functions as follows:

- TDSETLOG to turn tracing on or off and to change the name of the trace log
- TDINFLOG to determine whether tracing is enabled and to name the trace log
- TDSETSPT, TDLSTSPT, and TDINFSPT to enable, disable, and retrieve information about specific tracing
- TDWRTLOG to write your own record or add a system entry to the trace log file

For complete descriptions and examples of these functions, see the appropriate Mainframe Connect Server Option *Programmer's Reference*. PL/1 and COBOL versions of this guide are available.

Trace log

Under IMS TM, the Gateway-Library trace facility writes header, data, and API tracing information to the IMS TM log. The same IMS TM log contains errors, tracing, and accounting, so each record needs to indicate the type of record it is. This information comes from TDCUSTOM.

Trace log layout

Note The structures of the trace log have the same layout in CICS as in IMS TM, although the log headers differ slightly.

Table 5-1: Trace log layout

Position	Field name	Field type	Field description
1-2	log-length	unsigned 2-byte integer	Record length. The total length of this record. (Maximum size is 640.)

Position	Field name	Field type	Field description
3	log-type	unsigned byte	Type of record. There are two types of records for the trace log: trace and accounting. The trace type can be an error record (TDS-ERR-REC; the value is 1) or a trace record (TDS-TRACE the value is 2). The accounting log type value is 0.
4	log-direction	unsigned byte	Communication state: Shows whether the mainframe is in send or receive mode.
5-68	log-key	unsigned byte (max. length = 30) unsigned byte (length=8) unsigned byte (length = 24) unsigned 2-byte integer	Structure containing the following: user_id: The server login ID of the client, from the login packet. trace_resid: Trace resource ID. In CICS, this keeps track of who is doing the tracing. datetime: Date and time the SNA conversation or TCP/IP session started (TDACCEPT was issued). uniquekey: Reserved for future use to ensure record has unique key.
69-82	log-txp-name	unsigned byte	Transaction name.
82-112	log-conn-id	unsigned byte	Connection ID. Name by which the connection is known to TRS.
113	log-connp	unsigned byte	Name of associated TDPROC structure.
114-117	log-error-rc	4-byte integer	Value returned to the RETURN-CODE parameter of a Server Option function (TDS-xxxx). See the Mainframe Connect Client Option and Server Option *Messages and Codes* for more information about return codes.
118-121	log-err-type	4-byte integer	Type of error detected.
122-123	log-err-reserved	2-byte integer	Reserved for future use.
124-125	log-data-length	2-byte integer	Length of the data to be logged.
126-637	log-data	unsigned byte	Data, including the packet header and the data. For TDWRTLOG, this is the message being written to the log.
637-640	log_filler	unsigned byte (length=36)	Filler, to fill out record to 640 bytes.

Using the tracing facility

This section describes how to use the Server Option tracing facility. It contains the following subsections:

- Procedure for using the trace facility
- Trace table for individual transactions
- Specific tracing example

Procedure for using the trace facility

1 Call TDSETLOG and perform the following steps for global or specific tracing.

For this type of tracing	Do this
Global	a Set the trace flag to the TRACE ALL RPCS option. b Set the flag for each desired kind of tracing to TRUE. **Note** If you want to enable tracing for the entire program, TDSETLOG must precede TDACCEPT.
Specific	a Set the trace flag to the TRACE SPECIFIC RPCS option. b Set the flag for each desired type of tracing to TRUE.
Both global and specific	a Perform the previous steps for global and specific tracing. b If you are developing under CICS and want to enable API tracing, give the auxiliary trace log a CICS ID.

Note You can use TDINFLOG at any time to check the value of the settings.

2 For each transaction for which you want trace activity, call TDSETSPT and perform the following steps.

a Identify the transaction.

b Set the transaction trace flag to TRUE.

c Set the trace options flags for the type(s) of tracing desired.

Note You can enable tracing for up to eight transactions at a time.

Trace table for individual transactions

When you enable tracing for an individual transaction, TDSETSPT adds the transaction to a *trace table*. The trace table can contain up to eight entries. For examples of trace tables, see the next section, “Specific tracing example” on page 35.

When you disable tracing for a transaction, its position in the trace table becomes available for another transaction. If all eight positions are in use, you can trace more transactions only if you turn tracing off for one of the transactions in the list or set global tracing on.

You can query the trace table in two ways:

- Call TDINFSPT to determine if tracing is enabled for a specific transaction. You specify the transaction ID, and TDINFSPT returns the trace flag setting.
- Call TDLSTSPT to get a list of all transactions for which tracing is currently enabled. TDLSTSPT returns this list as an array.

Specific tracing example

The following example shows how to enable or disable tracing for specific transactions. It also shows how TDSETSPT calls affect the contents of the trace table. TDS packet tracing is initially turned on for eight specific transactions. Tracing continues for the specified functions until a TDSETSPT call turns tracing off for those functions, or until TDSETLOG disables tracing entirely.

This example does not show exact syntax or arguments; it merely indicates which flags and transactions are set. See the sample program in the appropriate *Programmer’s Reference* for the Server Option for an example of exact coding. PL/1 and COBOL versions of this guide are available.

```
*--------------------------------------------------------------------*
 *  First, initialize your environment and set on specific tracing. *
 *--------------------------------------------------------------------*
 CALL ’TDINIT’ ...
 CALL ’TDSETLOG’ ... (global flag: OFF,
       API flag: ON,
       header flag: OFF,
       data flag: OFF)...

 *-------------------------------------------------------------------*
 * Enable packet tracing (option 01) for a specific transaction.*
 *-------------------------------------------------------------------*
 CALL ’TDSETSPT’ ... (trace flag: ON,
      trace option: 01,
      tran ID: MYT1)...

*-------------------------------------------------------------------*
* Use the same parameter values except the transaction ID*
* in the next seven TDSETSPT calls.*
*-------------------------------------------------------------------*
CALL 'TDSETSPT' ... (tran ID: MYT2)...
CALL 'TDSETSPT' ... (tran ID: MYT3)...
CALL 'TDSETSPT' ... (tran ID: MYT4)...
CALL 'TDSETSPT' ... (tran ID: MYT5)...
CALL 'TDSETSPT' ... (tran ID: MYT6)...
CALL 'TDSETSPT' ... (tran ID: MYT7)...
CALL 'TDSETSPT' ... (tran ID: MYT8)...

*------------------------------------------------------------------*
* With tracing on, begin to accept and process client requests.*
*------------------------------------------------------------------*
CALL 'TDACCEPT'
.
.
.
```

At this point, the trace table looks like this:

Table 5-2: Sample trace table (1)

Transaction ID	Tracing flag
MYT1	TRUE
MYT2	TRUE
MYT3	TRUE
MYT4	TRUE
MYT5	TRUE
MYT6	TRUE
MYT7	TRUE
MYT8	TRUE

Later, you decide to turn on tracing for one more transaction, MYT9:

```
*-------------------------------------------------------------------*
* Try to turn on packet tracing for MYT9.                           *
*-------------------------------------------------------------------*
CALL 'TDSETSPT' ... (trace flag: ON,
          trace option: 01,
          tran ID: MYT9)...
*-------------------------------------------------------------------*
```

```
* The operation fails, and you get a return code of SOS,            *
* indicating that the trace table is full.                          *
* The contents of the trace table do not change.                    *
* To make room in the table for MYT9, you decide to                 *
* turn tracing off for MYT0.                                        *
*-------------------------------------------------------------------*
CALL 'TDSETSPT'...(trace flag: OFF,
           trace option: 01,
           tran ID: MYT0)...
*-------------------------------------------------------------------*
* The operation fails, and you get a return code                    *
* of ENTRY NOT FOUND, indicating that there is no such              *
* transaction listed in the trace table.                            *
* The contents of the trace table do not change.                    *
*-------------------------------------------------------------------*
* Since you apparently don‘t have an up-to-date list of the         *
* contents of the trace table, you use TDLSTSPT to survey           *
* all entries.                                                      *
*                                                                   *
* TDLSTSPT returns an array containing eight elements, each         *
* containing the transaction ID of an entry in the trace table      *
* for which tracing is TRUE.                                        *
*-------------------------------------------------------------------*
CALL 'TDLSTSPT' ...
*-------------------------------------------------------------------*
* You decide to turn tracing off for MYT3.                          *
*-------------------------------------------------------------------*
CALL 'TDSETSPT' ...(trace flag: OFF,
           trace option: 08,
           tran ID: MYT3)...
*-------------------------------------------------------------------*
* The operation succeeds;  the return code is OK.                   *
*-------------------------------------------------------------------*
```

The trace table now looks like this:

Table 5-3: Sample trace table (2)

Transaction ID	Tracing flag
MYT1	TRUE
MYT2	TRUE
MYT3	FALSE
MYT4	TRUE
MYT5	TRUE
MYT6	TRUE
MYT7	TRUE
MYT8	TRUE

Note The third position in the trace table is now considered empty.

When you try again to turn tracing on for MYT9, TDSETSPT moves it into the open position in the trace table.

```
*-------------------------------------------------------------------*
 * Try to enable tracing for MYT9.                                   *
 *-------------------------------------------------------------------*
 CALL ’TDSETSPT’ ...(trace flag: ON,
            trace option: 01,
            tran ID: MYT9)
 .
 .
 .
```

The trace table now looks like this:

Table 5-4: Sample trace table (3)

Transaction ID	Tracing flag
MYT1	TRUE
MYT2	TRUE
MYT9	TRUE
MYT4	TRUE
MYT5	TRUE
MYT6	TRUE
MYT7	TRUE
MYT8	TRUE

Still later, you decide to turn on tracing for MYT2:

```
*-------------------------------------------------------------------*
 *-------------------------------------------------------------------*
 * Try to enable tracing for MYT2.                                   *
 *-------------------------------------------------------------------*
 CALL ’TDSETSPT’ ... (trace flag: OFF,
            trace option: 01,
            tran ID: MYT2)...
 *-------------------------------------------------------------------*
 * The operation fails. You get a TDS DUPLICATE ENTRY return code, as*
 * tracing is already enabled for the transaction–no action needed.  *
 *-------------------------------------------------------------------*
```

Accounting

The Server Option allows you to record accounting information at the mainframe and at TRS. Mainframe-based accounting is independent of TRS-based accounting. For example, when the TRS accounting facility records packet is received, it is recording the number of packets sent from the mainframe to TRS; when the mainframe accounting facility records packet is received, it is recording the number of packets sent from TRS to the mainframe.

Accounting can be enabled at TRS, at the mainframe, or both. For information on TRS accounting, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*. This section describes accounting at the mainframe.

Note The mainframe accounting facility uses elapsed time.

Under MVS, a point in time is the number of seconds and milliseconds since 00:00:00 on 1/1/70. Elapsed time is the difference between the ending time and the starting time, that is, the number of seconds and milliseconds that elapse between the start and end points.

The following subsections explain the Server Option functions and the accounting log:

- The Server Option accounting functions
- Accounting log

The Server Option accounting functions

To enable mainframe server accounting information, call TDSETACT in your Server Option program. TDSETACT begins recording when your program issues a TDACCEPT and continues until the program issues TDFREE. Use TDINFACT to learn whether accounting recording is enabled and the name of the accounting log file.

See the *Programmer's Reference* for the Server Option for complete descriptions and examples of these functions. PL/1 and COBOL versions of this guide are available.

Accounting log

Under IMS TM, mainframe-based accounting stores accounting information in the IMS TM log.

Warning! The IMS TM log must be set up during customization using IMSLOGTYPE.

Accounting log layout

Note The structures of the accounting log have the same layout in CICS as in IMS TM, although the log headers differ slightly.

Table 5-5: Accounting log layout

Position	Field name	Field type	Field description
1-2	acct-length	unsigned 2-byte integer	Record length. The total length of this accounting record. (Maximum size of a CICS record is 256.)
3	acct-type	unsigned byte	Type of record. For the accounting log, this type is always TDS-ACCT-REC.
4	acct-direction	unsigned byte	Reserved for future use.
5-68			Structure containing the following:
	acct-key	unsigned byte (max. length = 30)	user_id: Client's server login ID, from the login packet.
		unsigned byte (length = 24)	trace_resid: Trace resource ID. In CICS, this keeps track of who is doing the tracing.
		unsigned byte (length=8)	datetime: Date and time the SNA conversation or TCP/IP session started. (TDACCEPT was issued.)

Position	Field name	Field type	Field description
		unsigned 2-byte integer	uniquekey: Reserved for future use to ensure record has unique key.
69-82	acct-txp-name	unsigned byte	Transaction name.
83-112	acct-server-id	unsigned byte	TRS name. Name of the TRS sending the current request.
113-142	acct-conn-id	unsigned byte	Connection ID. Name by which the connection is known to TRS.
143-144	Filler	unsigned byte	Filler to allow next entries to be fullwords.
145-148	acct-tot-secs-wall	4-byte integer	Elapsed wall clock time, in seconds, during the SNA conversation or TCP/IP session.
149-152	acct-tot-fracsecs-wall	unsigned 4-byte integer	Elapsed wall clock time, in milliseconds during the SNA conversation or TCP/IP session.
153-156	acct-tot-secs-cpu	4-byte integer	CPU time used, in seconds during the SNA conversation or TCP/IP session.
157-160	acct-tot-fracsecs-cpu	unsigned 4-byte integer	CPU time used, in milliseconds, during the SNA conversation or TCP/IP session.
161-164	acct-tot-sent-bytes	4-byte integer	Total number of TDS bytes sent during an SNA conversation or TCP/IP session.
165-168	acct-tot-sent-packets	4-byte integer	Total number of TDS packets sent during an SNA conversation or TCP/IP session.
169-172	acct-tot-sent-msgs	4-byte integer	Total number of TDS messages sent during an SNA conversation or TCP/IP session.
173-176	acct-tot-sent-rows	4-byte integer	Total number of TDS rows sent during an SNA conversation or TCP/IP session.
177-180	acct-tot-sent-requests	4-byte integer	Total number of RPCs or SQL requests sent during an SNA conversation or TCP/IP session. For the Server Option, this is always 0.
181-184	acct-tot-rcvd-bytes	4-byte integer	Total number of TDS bytes received during an SNA conversation or TCP/IP session.
185-188	acct-tot-rcvd-packets	4-byte integer	Total number of TDS packets received during an SNA conversation or TCP/IP session.

Position	Field name	Field type	Field description
189-192	acct-tot-rcvd-msgs	4-byte integer	Total number of TDS messages received during an SNA conversation or TCP/IP session.
193-196	acct-tot-rcvd-rows	4-byte integer	Total number of TDS rows received during an SNA conversation or TCP/IP session. For the Server Option, this is always 0.
197-200	acct-tot-rcvd-requests	4-byte integer	Total number of RPCs or SQL requests received during an SNA conversation or TCP/IP session.
201-204	acct-tot-rcvd-cancels	4-byte integer	Total number of Cancels or Attentions received during an SNA conversation or TCP/IP session.
205-208	acct-reserved1	4-byte integer	Reserved for future use.
209-212	acct-reserved2	4-byte integer	Reserved for future use.
213-216	acct-reserved3	4-byte integer	Reserved for future use.
217-220	acct-reserved4	4-byte integer	Reserved for future use.
221-236	acct_fill	unsigned byte (length=36)	Filler, to fill out record to 256 bytes.

APPENDIX A

Customization Options

This appendix includes the following topics:

Overview

You can customize Sybase mainframe access components to meet the requirements at your site. The customization load module SYGWXCPH is a table created by assembling and linking five macros:

- SYGWMCST is a global customization macro.
- SYGWMCXL is a character set conversion macro.
- SYGWDRIV specifies which dynamic network driver(s) are used at the site.
- SYGWHOST provides mapping between Sybase Server names and TCP/IP addresses or host names.
- SYGWLKEY is a license key macro.

The SYGWXCPH table is shared by the Client Option and the Server Option.

Customizing global options (SYGWMCST)

SYGWMCST, one of the macros in table SYGWXCPH, provides options for customizing the Client Option and the Server Option. Some Server Option parameters are used only for customizing the DB2 UDB Option for CICS. You can customize SYGWMCST using the provided JCL member.

The following table describes SYGWMCST parameters. These apply to both the Client Option and Server Option, except where noted.

Table A-1: Complete list of SYGWMCST parameters

Parameter	Default	Format	Purpose
ACCESSCODE (Server Option only)	blank	Up to 32 characters	Defines an access code, which is then compared to the access code supplied by Server Option programs using TDGETUSR. If the access codes do not match, the client password is not returned to the caller of Server Option programs using TDGETUSR. See the appropriate Mainframe Connect Server Option *Programmer's Reference* for details on TDGETUSR.
ACCESSCODESW (Server Option only)	N	Y or N	Turns on/off access code comparison (see *ACCESSCODE* value). When *ACCESSCODESW*=N (default), the logged-in password is always returned to the caller of Server Option programs using TDGETUSR. When *ACCESSCODESW*=Y, the logged-in password is returned only if the access code passed to TDGETUSR matches the access code specified in SYGWMCST *ACCESSCODE*.
CHARSETSRV	iso_1	Up to 32 characters	Specifies the default character set that the Client Option or Server Option uses internally. The valid values are iso_1 and utf8. **Note** The value utf8 is valid only if *USEIBMUNICODE* is set to Y.
DEBUGSW	N	Y or N	Specifies whether or not debugging messages, used in troubleshooting, should be displayed in the system log.

Parameter	**Default**	**Format**	**Purpose**
DECPOINT *(Server Option only)*	’.’ (decimal point)	Either a decimal point or comma delimited by single quotation marks	Decimal point indicator. For use only with the DB2 UDB Option for CICS.
DEFLTPROTOCOL	TCP	TCP	Specifies the default network driver protocol.
DQUOTETRAN *(Server Option only)*	Y	Y or N	Used only with the DB2 UDB Option for CICS. Make this setting consistent with your DB2 configuration. When *DQUOTETRAN*=Y (default), double quotes are translated to single quotes in incoming SQL text. If you are using an ODBC driver, set *DQUOTETRAN*=N. **Note** If you are using double-byte or multi-byte characters for DB2 metadata, set *DQUOTETRAN*=N.
IMSLOGTYPE *(IMS TM only)*	A1	A value greater than or equal to A0.	Specifies a log type. IMS TM reserves values less than A0.
LONGVARTRUNC	N	Y or N	Indicates whether to truncate *LongVarChar* and *VarBinary*. *For CICS only:* Coordinate this setting with the DirectConnect for z/OS Option TRS. If either this parameter or the TRS *TruncateLV* configuration property is set for truncation, truncation occurs. If you do not want truncation, set this parameter to N and make sure the TRS *TruncateLV* configuration property is set to No. See the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

Parameter	Default	Format	Purpose
MVSDDNAME *(IMS TM and MVS only)*	blank	From 1 to 8 characters	The DD name of the MVS Open Client/Open Server log file. If this parameter is left blank (the default), MVS transactions are not logged. If you enter a DD name of 1-8 characters, MVS transactions are logged. The name specified here must match a DD name specified in each MVS transaction profile job. *MVSDDNAME* must match a DD name specified in the JCL for one of the following: • an MVS job • an MVS started task • the MVS transaction profile (if run in an APPC initiator as a transaction)
NATLANGUAGESRV	us_english	Up to 32 characters	Designates the default national language used by the Client Option or Server Option. Also see the *CHARSETSRV* property.
ROWLIMIT *(Server Option only)*	0 (zero)		Used only by the DB2 UDB Option for CICS. When *ROWLIMIT*=0, there is no limit to the number of rows that can be sent. *ROWLIMIT*=*n*, where *n*=a number of rows, indicates the global limit of rows that can be sent.
USEIBMUNICODE	N	Y or N	Specifies whether or not Unicode support for a particular z/OS installation is enabled through the IBM conversion environment and services. • If *USEIBMUNICODE*=Y, IBM support is used for character set conversions. • If *USEIBMUNICODE*=N, conversion is accomplished through the product-supplied translation tables. See "Using the IBM z/OS conversion environment and services."

Note The following parameters are no longer used:

- *DBCS*
- *NOUDTTRAN*
- *PARSEXITNAME*
- *PARSEXITSW*

Using the IBM z/OS conversion environment and services

Unicode support in the Client Option and Server Option is based on Unicode support provided by IBM z/OS, including the conversion environment and conversion services. With the conversion environment and services installed and set up, the Client Option and Server Option can convert character streams from one Coded Character Set Identifier (CCSID) to another. This functionality is provided in addition to the support for language and character sets offered in previous versions.

To install IBM Unicode support, use the following procedure.

❖ Installing IBM Unicode support

1 Create an *IMAGE* member in *SYS1.PARMLIB* using the CUNMIUTL utility.

2 Copy the *CUNIMG01* member from *WORK.IMAGE* to *SYS1.PARMLIB*. The *CUNIMG01* member is loaded into z/OS using the following command:

```
SET UNI=01
```

The following command displays the current active image and the character set conversions defined for that image:

```
DISPLAY UNI, ALL
```

To enable Unicode support for the Client Option and Server Option, set the *USEIBMUNICODE* configuration parameter to Y. The *USEIBMUNICODE* parameter is specified in the SYGWMCST macro in the SYGWXCPH customization module. For more information on installing Unicode support for IBM z/OS, see "Support for Unicode Using Conversion Services" (SA22-7649-01).

Customizing mainframe character set conversion options (SYGWMCXL)

SYGWMCXL is the character set conversion macro in the table SYGWXCPH. The following considerations apply in using the SYGWMCXL macro:

- When Unicode support is disabled (*USEIBMUNICODE*=N) and the original translation method is used, SYGWMCXL can be used to override supplied SBCS translation tables or to define new SBCS translation tables.
- When Unicode support is enabled (*USEIBMUNICODE*=Y), SYGWMCXL is used to create definition entries for the character sets to be used in the Client Option or Server Option conversions at a particular installation. These entries are created in addition to system-generated entries.

Note All EBCDIC-to-ASCII and ASCII-to-EBCDIC translation for Client Option or the Server Option occurs on the mainframe.

Overriding the supplied SBCS translation tables

For SBCS, shipped character sets are called *predefined*, and the character sets you define are called *user-definable*.

Predefined character sets

Predefined SBCSs shipped with the product include:

SBCS	Definition
ascii_8	Default used for logins and for IBM cp1027 (code page 1027) support
cp437 (code page 437)	Used by IBM PCs
cp850 (code page 850)	IBM/Microsoft Multilingual Character Set, used by IBM PCs
iso_1 (ascii 0819)	International ISO standard, 8-bit character set for many systems, and the default for Adaptive Server Enterprise on several platforms
mac (Macintosh Roman)	Default used by Macintosh systems
roman8	Default Hewlett-Packard proprietary character set

Note Unpredictable failures can occur if the character set names are changed from lowercase to uppercase.

User-defined character sets

You can change all attributes for user character sets. The SBCS settings of the parameters for SYGWMCXL are:

Table A-2: SYGWMCXL parameters for SBCS

Parameter	Value
A2E	Optional ASCII-to-EBCDIC translate overrides
E2A	Optional EBCDIC-to-ASCII translate overrides
CHARSET	Name of the SBCS
CHARSETBYTES	S for SBCS
TYPE	Valid types: • INITIAL • ENTRY (default) • FINAL

If there is no override entry for a predefined character set, a default entry is generated with the appropriate translation tables and other attributes for that character set. A total of 99 character sets, including custom-generated character set entries, is supported.

The minimum translate customization entries are:

```
SYGWMCXL TYPE=INITIAL
SYGWMCXL TYPE=FINAL
```

These entries generate all of the predefined SBCSs.

Defining new SBCS translation tables

For SBCSs, you can modify the translation tables shipped with the product and create new translation tables with names you define.

Warning! Do not use the shipped table names for the tables you create.

If you create new tables, it is important to coordinate with the person responsible for the Sybase client. The client uses the names of the tables you create to issue logins to the DirectConnect for z/OS Option TRS.

When you finish customizing the SBCS translation tables, rebuild the SYGWXCPH module, and load the new module for your revisions to take effect. Instructions are provided in “Building a global customization module (SYGWXCPH)” on page 57.

Overriding defaults and creating new tables on the mainframe

The SYGWMCXL macro generates translation tables to convert between ASCII and EBCDIC character sets. Default translation tables are generated for the following ASCII character sets:

- ascii_8
- cp437
- cp850
- iso_1
- mac
- roman8

Note Unpredictable failures can occur if the character set names are changed from lowercase to uppercase.

These default tables also provide the "base" for any character set changes or new tables you want to define. For details on the base translate tables, see Appendix B, “Translation Tables.”

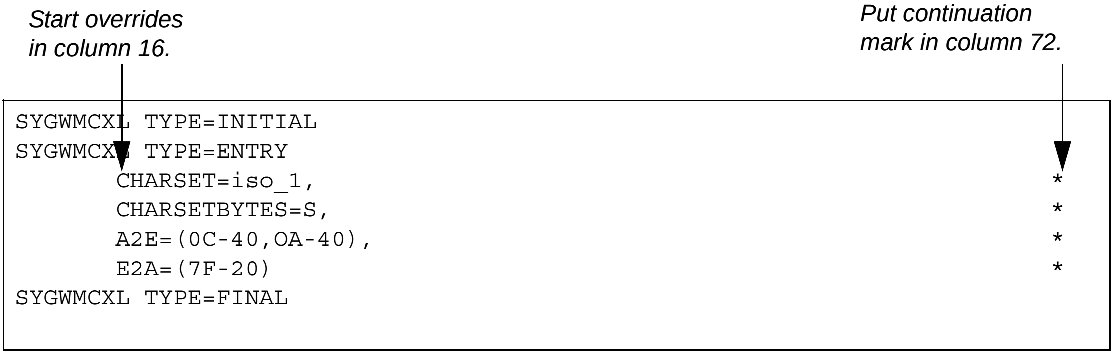
You can change all attributes for user character sets. An entry is added to the translate table, specifying the appropriate character set attributes. Two examples follow for overriding defaults.

The first example, Figure A-1, shows how to use A2E and E2A macro parameters to override the ASCII-to-EBCDIC defaults. You can use uppercase or lowercase to define the parameters.

When you override the ASCII-to-EBCDIC defaults, the appropriate base table is picked up as a template for the character overrides or user-defined character sets, thus generating a default table. In the following example, the client is using us_english, which is not predefined.

The second example shows how to modify the default character set, iso_1, for Hebrew, creating a new table:

Figure A-1: Using A2E and E2A example



Start overrides in column 16.

Put continuation mark in column 72.

```
SYGWMCXL TYPE=INITIAL
SYGWMCXL TYPE=ENTRY
       CHARSET=iso_1,                                                   *
       CHARSETBYTES=S,                                                  *
       A2E=(0C-40,OA-40),                                               *
       E2A=(7F-20)                                                      *
SYGWMCXL TYPE=FINAL
```

This example converts both of the following:

- ASCII form feeds (x'0C') and line feeds (x'0A') to EBCDIC spaces (x'40')
- EBCDIC DELs (x'7F') to ASCII space (x'20')

Creating a new table

The following example shows how to modify the default character set, iso_1, for Hebrew, creating a new table:

```
* These SYGWMCXL macro calls modify the iso_1 character set
* to Hebrew.
*
  SYGWMCXL        TYPE=INITIAL
  SYGWMCXL TYPE=ENTRY,                       *
     CHARSET=(unique_name),                  *
     CHARSETBYTES=S,                         *
     A2E=(E0-41,E1-42,E2-43,E3-44,E4-45,E5-46,E6-47,E7-48,E8-*
     49,E9-51,EA-52,EB-53,EC-54,ED-55,EE-56,EF-57,F0-58,F1-59*
     ,F2-62,F3-63,F4-64,F5-65,F6-66,F7-67,F8-68,F9-69,FA-71),*
     E2A=(41-E0,42-E1,43-E2,44-E3,45-E4,46-E5,47-E6,48-E7,49-*
     E8,51-E9,52-EA,53-EB,54-EC,55-ED,56-EE,47-EF,58-F0,59-F1*
     ,62-F2,63-F3,64-F4,65-F5,66-F6,67-F7,68-F8,69-F9,71-FA)
  SYGWMCXL TYPE=FINAL
*
* Assembler END is required.
*
END
```

For the *CHARSET* parameter, specify a unique name. This generates a new user-defined table. Provide the name to the appropriate person at the Sybase client site. The client login packet uses this name.

Defining new character set entries

In using the IBM Unicode conversion environment and services, the SYGWMCXL macro is used to create definition entries for all the character sets that will be used at a particular site and that are not already defined as system character sets. Table A-3 describes the parameters used in the SYGWMCXL macro to create a definition entry:

Table A-3: SYGWMCXL macro parameters

Parameter	Value
CHARSET	The name of the SBCS or DBCS character set.
CHARSET BYTES	An S to denote SBCS, or a D to denote DBCS.
CCSID	The CCSID for the character set.
CHARSETYPE	The type of character set. A denotes ASCII, and E denotes EBCDIC.
CHARSIZE	The maximum length of a character, from 1 to 4 bytes.
PAD	The padding character. The value of this parameter depends on the character set type. For ASCII, the padding character is 20. For EBCDIC, the padding character is 40.

The following examples illustrate definitions for Russian and Japanese EBCDIC character sets, which are code pages 1025 and 939, respectively.

Example: code page 1025

```
SYGWMCXL TYPE=ENTRY,
      CHARSET=Russian,CHARSETBYTES=S,
      CCSID=1025,CHARTYPE=E,CHARSIZE=1,PAD=40
```

Example: code page 939

```
SYGWMCXL TYPE=ENTRY,
      CHARSET=cp939,CHARSETBYTES=D,
      CCSID=939,CHARTYPE=E,CHARSIZE=2,PAD=40
```

In addition to the default ASCII SBCS translation tables, the following names are used to generate system entries for ASCII DBCS character sets:

- *sjis* – Japanese code page cp943 or cp932
- *eucjis* – Japanese code page cp33722
- *cp950* – traditional Chinese Big5 or CP950
- *cp936* – simplified Chinese GBK or cp936

If you use any of these names, you do not need to create a new definition.

Customizing dynamic network drivers (SYGWDRIV)

SYGWDRIV, a macro in the SYGWXCPH table, defines the dynamic network drivers for the Client Option or Server Option.

Note If you are using a TCP/IP driver, you must also configure the SYGWHOST macro.

CICS network drivers

The following default drivers are shipped with the Client Option or Server Option, depending on the environment:

Table A-4: CICS network drivers

Driver	Load module name	Comments
LU 6.2	LU62CICS	Uses CICS LU 6.2 API
IBM TCP/IP	TCPCICS	Uses IBM EZACICAL API
CPIC	CPICCICS	Uses CICS CPIC Support

The CICS JCL member *IxHOST* contains the following macro definitions, which set up support for all four network drivers:

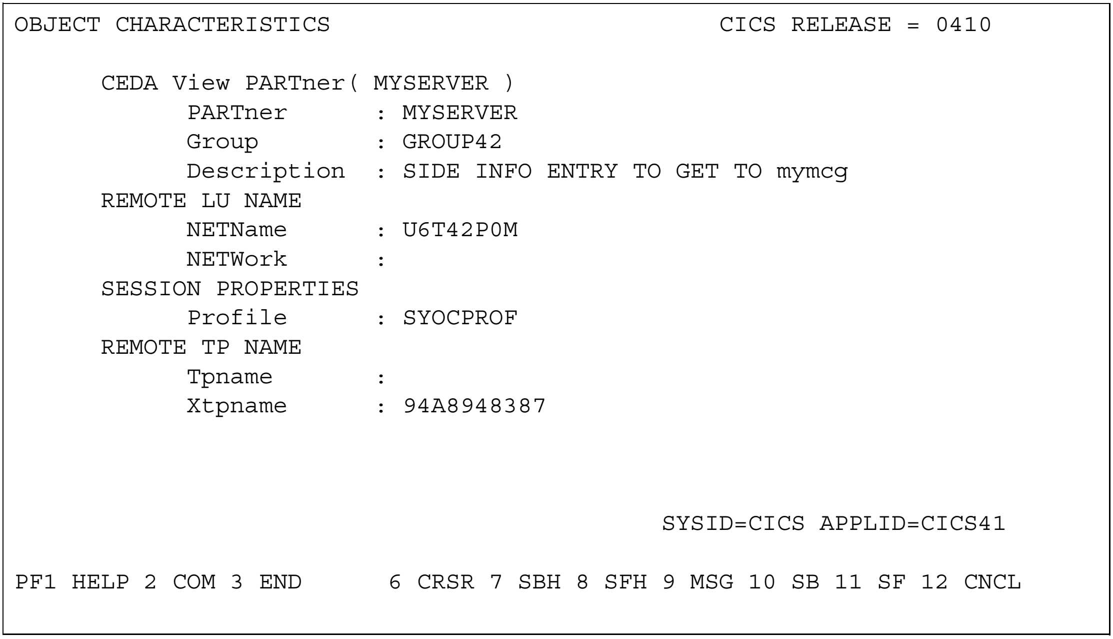
```
        SYGWDRIV TYPE=INITIAL
*
        SYGWDRIV TYPE=ENTRY,ENV=CICS,NETD=LU62
        SYGWDRIV TYPE=ENTRY,ENV=CICS,NETD=CPIC
        SYGWDRIV TYPE=ENTRY,ENV=CICS,NETD=TCP
*
        SYGWDRIV TYPE=FINAL
```

CPI-C CICS network driver

If you use the CPI-C CICS driver, you must use CEDA to define an entry in the CICS PARTNER Table. Due to an IBM requirement, each Partner entry must be exactly 8 characters in length and use A-Z, 0-9. If your actual server name is not 8 characters, put an alias for it in your *interfaces* file.

For example:

Figure A-2: CEDA panel



```
OBJECT CHARACTERISTICS                           CICS RELEASE = 0410

      CEDA View PARTner( MYSERVER )
            PARTner      : MYSERVER
            Group        : GROUP42
            Description  : SIDE INFO ENTRY TO GET TO mymcg
      REMOTE LU NAME
            NETName      : U6T42P0M
            NETWork      :
      SESSION PROPERTIES
            Profile      : SYOCPROF
      REMOTE TP NAME
            Tpname       :
            Xtpname      : 94A8948387



                                             SYSID=CICS APPLID=CICS41

PF1 HELP 2 COM 3 END      6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Enter the PARTner and Remote TP name field values as follows:

- PARTner—This must be *exactly* 8 characters long. An alias for the eight-character name should be added to the *interfaces* file if necessary.
- Remote TP name—If the name of your server is in uppercase, enter it in the Tpname field. If the name of your server is in lowercase, enter the EBCDIC hexadecimal name in the Xtpname field.

Note If you enter a lowercase name in the Tpname field, CEDA changes it to uppercase and an erroneous entry is passed.

Customizing the TCP/IP driver (SYGWHOST)

The SYGWHOST macro is part of the SYGWXCPH global customization module. This macro is used only for the Client Option in connections from the mainframe to other applications. It is required only if you are using a TCP/IP driver, in which case you must configure SYGWHOST to define the mapping between Sybase server names and TCP/IP addresses or hostnames. Do not depend on the default shipped with the installation to work in your environment.

For the Server Option, only the TYPE=INITIAL and TYPE=FINAL macros are required. The TYPE=ENTRY macros are required only for the Client Option.

This section documents the following topics:

- Macro parameters
- Macro formats

Macro parameters

There are six parameters in this macro:

Parameter	Definition
HOSTNAME	The name of the host on which the Sybase server resides. The maximum length of the host name is 24 characters. If a value is provided for the *IPADDR* parameter, the *HOSTNAME* parameter is ignored, and no DNS search is performed.
IBMTCPADDRSPACE	Designates the name of the IBM TCP/IP address space. This parameter can be specified as either of the following: • A hard coded value of up to 8 characters. • A system symbolic name. System symbolic names are defined in the IEASYMxx PARMLIB member and are limited to 7 characters preceded by "&&". For example, the symbolic name "SYBTCP" would be designated as follows: `IBMTCPADDRSPACE=&&SYBTCP` Symbolic names allow the use of a common SYGWXCPH configuration module across multiple LPARs, even if each LPAR has a different TCP address space name. The default address space name is TCPIP.
IPADDR	The IP address of the host on which the Sybase server resides. If a value is provided for this parameter, the *HOSTNAME* parameter is ignored.

Parameter	Definition
LISTENER	One of the following: • *LAN* if the listen port is for a LAN-based server (default) • *CICS* if the listen port is for an CICS Server Option listener • *IMS* if the listen port is for an IMS TM Server Option listener
LSTNPORT	The listen port of the server specified by *SERVERNAME*.
SERVERNAME	The 1-30 byte name of a Sybase server.

Macro formats

There are three macro formats: TYPE=INITIAL, TYPE=ENTRY, and TYPE=FINAL.

TYPE=INITIAL

The format of TYPE=INITIAL is:

```
SYGWHOST TYPE=INITIAL
```

TYPE=ENTRY

The format of TYPE=ENTRY is:

```
SYGWHOST TYPE=ENTRY
    IBMTCPADRSPCNAME=&&TCP,
    LISTENER=(LAN,CICS,IMS)
    LSTNPORT=99999,
    SERVERNAME=sybase10,
    HOSTNAME=myhost
```

TYPE=FINAL

The format of TYPE=FINAL is:

```
SYGWHOST TYPE=FINAL
```

Defining license keys (SYGWLKEY)

The SYGWLKEY macro is part of the SYGWXCPH global customization module. It is used to define the customer license key that is verified at run time.

There are two parameters in this macro:

Parameter	Definition
PRODUCT	The product related to the license key, either the Client Option or the Server Option.
KEY	Defines the license key given for a product. The license key is a 22-character numeric value.

The following is an example of SYGWLKEY.

```
SYGWLKEY TYPE=INITIAL
SYGWLKEY TYPE=ENTRY,PRODUCT=OCC,KEY=19320-00000-20$*#-#19$B
SYGWLKEY TYPE=ENTRY,PRODUCT=OSC,KEY=19300-00000-00E2G-4K##6
SYGWLKEY TYPE=FINAL
```

Note For the Client Option and Server Option for CICS, the license keys are kept in a VSAM file. Run the generated job *IxLIC* to install these.

Building a global customization module (SYGWXCPH)

The install process in Chapter 3, “Installation and Configuration” creates the the *IxTCP* job (where *x* is an integer that denotes the order in which the job is to be run in the overall sequence of jobs). The *IxTCP* job can be run to create a basic version of the SYGWXCPH global customization module, which contains the following macros:

- SYGWMCST
- SYGWMCXL
- SYGWDRIV
- SYGWHOST
- SYGWLKEY

The SYGWLKEY macro is for IMS or MVS and a relocated object module, TDSGLOB.

APPENDIX B

Translation Tables

This appendix includes the following topics:

Understanding the ASCII-EBCDIC and EBCDIC-ASCII translation tables

This appendix shows the default settings for the ASCII-EBCDIC and EBCDIC-ASCII translation tables before any user overrides.

Note The translation tables shown here are used in date conversion only if Unicode support is disabled and *USEIBMUNICODE*=N.

The four pairs of default, or "base," tables are:

- ASCII_8
- ISO_1
- cp437
- cp 850

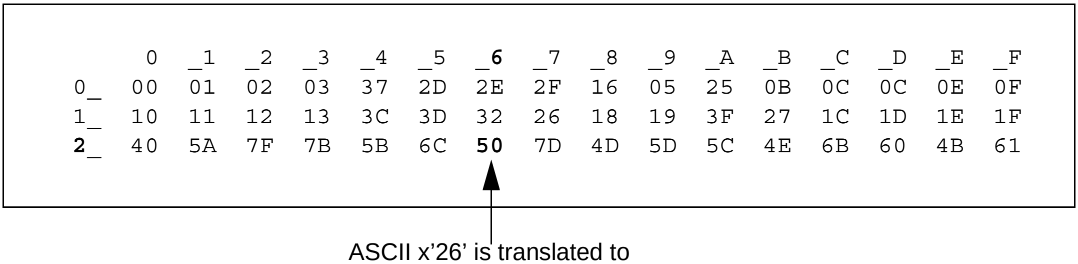
Each pair includes a table for ASCII-to-EBCDIC translation, and one for EBCDIC-to-ASCII translation.

Note As supplied, all ASCII character sets translate to and from EBCDIC code page 500 (iso_1) on the mainframe by default.

For the ASCII-to-EBCDIC tables, find the leftmost hexadecimal ASCII digit to the left of the table as a digit followed by an underscore. Find the rightmost hexadecimal ASCII digit on top of the table as a digit preceded by an underscore.

Here is an example from the default table in the section “ASCII_8, ASCII-to-EBCDIC translation table” on page 62.

Figure B-1: Example from the ASCII_8, ASCII-to-EBCDIC translation table



	0	_1	_2	_3	_4	_5	_**6**	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0C	0E	0F
1_	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
2_	40	5A	7F	7B	5B	6C	**50**	7D	4D	5D	5C	4E	6B	60	4B	61

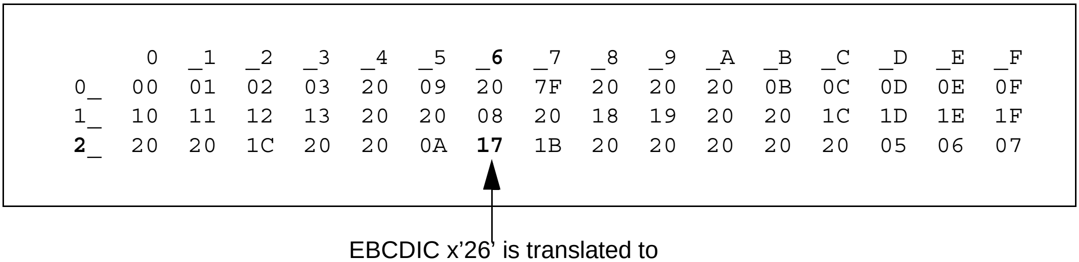
ASCII x’26’ is translated to
EBCDIC x’50’.

To locate ASCII x'26', find row 2_ to the left of the table, and proceed along that row to the column headed by _6. At the intersection is x'50'. Therefore, ASCII x'26' is translated to EBCDIC x'50'.

For the EBCDIC-to-ASCII tables, find the leftmost hexadecimal EBCDIC digit to the left of the table as a digit followed by an underscore. Find the rightmost hexadecimal EBCDIC digit on top of the table as a digit preceded by an underscore.

Here is an example from the default table in the section “ASCII_8, ASCII-to-EBCDIC translation table” on page 62.

Figure B-2: Example from the ASCII_8, EBCDIC-to-ASCII translation table



	0	_1	_2	_3	_4	_5	**_6**	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	20	09	20	7F	20	20	20	0B	0C	0D	0E	0F
1_	10	11	12	13	20	20	08	20	18	19	20	20	1C	1D	1E	1F
2_	20	20	1C	20	20	0A	**17**	1B	20	20	20	20	20	05	06	07

EBCDIC x’26’ is translated to ASCII x’17’.

To locate EBCDIC x'26', find row 2_on the left side of the table; then proceed along that row to the column headed by _6. At the intersection is x'17'. Therefore, EBCDIC x'26' is translated to ASCII x'17'.

Warning! If you create a new table from a default table, give the new table a unique name and coordinate with the appropriate person at the Sybase client site. The client can use the name to issue logins to TRS.

Default ASCII_8 translation tables

This section contains the following tables:

- ASCII_8, ASCII-to-EBCDIC translation table
- ASCII_8, EBCDIC-to-ASCII translation table

The ASCII-to-EBCDIC translation tables in this section are the base tables for the following predefined system SBCSs:

- ascii_8
- roman8
- mac
- ibmascii

Use these tables as the base ASCII-to-EBCDIC translation table for user-definable character sets.

ASCII_8, ASCII-to-EBCDIC translation table

Figure B-3: ASCII_8, ASCII-to-EBCDIC translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0D	0E	0F
1_	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
2_	40	5A	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3_	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4_	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5_	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	AD	E0	BD	5F	6D
6_	79	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7_	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	8B	6A	9B	A1	07
8_	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
9_	90	91	92	93	94	95	96	97	98	99	9A	4A	9C	9D	9E	9F
A_	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	5F	AB	AC	AD	AE	AF
B_	B0	B1	B2	4F	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BC
C_	AB	C1	C2	C3	BF	8F	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D_	D0	D1	D2	D3	D4	D5	D6	D7	D8	BB	AC	DB	DC	DD	DE	DF
E_	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F_	F0	9E	AE	8C	F4	F5	F6	F7	A1	AF	FA	FB	FC	FD	9F	FF

ASCII_8, EBCDIC-to-ASCII translation table

Figure B-4: ASCII_8, EBCDIC-to-ASCII translation table

	0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	20	09	20	7F	20	20	20	0B	0C	0D	0E	0F
1_	10	11	12	13	20	20	08	20	18	19	20	20	1C	1D	1E	1F
2_	20	20	1C	20	20	0A	17	1B	20	20	20	20	20	05	06	07
3_	20	20	16	20	20	20	20	04	20	20	20	20	14	15	20	1A
4_	20	20	20	20	20	20	20	20	20	20	9B	2E	3C	28	2B	B3
5_	26	20	20	20	20	20	20	20	20	20	21	24	2A	29	3B	AA
6_	2D	2F	20	20	20	20	20	20	20	20	7C	2C	25	5F	3E	3F
7_	20	20	20	20	20	20	20	20	20	60	3A	23	40	27	3D	22
8_	20	61	62	63	64	65	66	67	68	69	20	7B	F3	20	20	C5
9_	20	6A	6B	6C	6D	6E	6F	70	71	72	20	7D	20	20	F1	FE
A_	20	7E	73	74	75	76	77	78	79	7A	20	C0	DA	5B	F2	F9
B_	20	20	20	20	20	20	20	20	20	20	20	D9	BF	5D	20	C4
C_	7B	41	42	43	44	45	46	47	48	49	20	20	20	20	20	20
D_	7D	4A	4B	4C	4D	4E	4F	50	51	52	20	20	20	20	20	20
E_	5C	20	53	54	55	56	57	58	59	5A	20	20	20	20	20	20
F_	30	31	32	33	34	35	36	37	38	39	20	20	20	20	20	20

Default ISO_1 translation tables

This section contains the following tables:

- ISO_1 ASCII-to-EBCDIC translation table
- ISO_1 EBCDIC-to-ASCII translation table

The ASCII-to-EBCDIC translation tables in this section are the base table for the predefined system iso_1 character set.

ISO_1 ASCII-to-EBCDIC translation table

Figure B-5: ISO_1 ASCII-to-EBCDIC translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0D	0E	0F
1_	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
2_	40	4F	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3_	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4_	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5_	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	4A	E0	5A	5F	6D
6_	79	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7_	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	C0	BB	D0	A1	07
8_	20	21	22	23	24	15	06	17	28	29	2A	2B	2C	09	0A	1B
9_	30	31	1A	33	34	35	36	08	38	39	3A	3B	04	14	3E	FF
A_	41	AA	B0	B1	9F	B2	6A	B5	BD	B4	9A	6A	BA	CA	AF	BC
B_	90	8F	EA	FA	BE	A0	B6	B3	9A	DA	9B	8B	B7	C7	B9	AB
C_	64	65	62	66	63	67	9E	69	74	71	72	73	78	75	76	77
D_	AC	69	ED	EE	EB	EF	EC	BF	80	FD	FE	FB	FC	AD	AE	59
E_	44	45	42	46	43	47	9C	48	54	51	52	53	58	55	56	57
F_	8C	49	CD	CE	CB	CF	CC	E1	70	DD	DE	DB	DC	8D	8E	DF

ISO_1 EBCDIC-to-ASCII translation table

Figure B-6: ISO_1 EBCDIC-to-ASCII translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	9C	09	86	7F	97	8D	8E	0B	0C	0D	0E	0F
1_	10	11	12	13	9D	85	08	87	18	19	92	8F	1C	1D	1E	1F
2_	80	81	82	83	84	0A	17	1B	88	89	8A	8B	8C	05	06	07
3_	90	91	16	93	94	95	96	04	98	99	9A	9B	14	15	9E	1A
4_	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	5B	2E	3C	28	2B	21
5_	26	E9	EA	EB	E8	ED	EE	EF	EC	DF	5D	24	2A	29	3B	5E
6_	2D	2F	C2	C4	C0	C1	C3	C5	C7	D1	A6	2C	25	5F	3E	3F
7_	F8	C9	CA	CB	C8	CD	CE	CF	CC	60	3A	23	40	27	3D	22
8_	D8	61	62	63	64	65	66	67	68	69	AB	BB	F0	FD	FE	B1
9_	B0	6A	6B	6C	6D	6E	6F	70	71	72	AA	BA	E6	B8	C6	A4
A_	B5	7E	73	74	75	76	77	78	79	7A	A1	BF	D0	DD	DE	AE
B_	A2	A3	A5	B7	A9	A7	B6	BC	BD	BE	AC	7C	AF	A8	B4	D7
C_	7B	41	42	43	44	45	46	47	48	49	AD	F4	F6	F2	F3	F5
D_	7D	4A	4B	4C	4D	4E	4F	50	51	52	B9	FB	FC	F9	FA	FF
E_	5C	F7	53	54	55	56	57	58	59	5A	B2	D4	D6	D2	D3	D5
F_	30	31	32	33	34	35	36	37	38	39	B3	DB	DC	D9	DA	9F

Default cp437 (code page 437) translation tables

This section contains the following tables:

- cp437 ASCII-to-EBCDIC translation table
- cp437 EBCDIC-to-ASCII translation table

The ASCII-to-EBCDIC translation tables in this section are the base tables for the predefined system cp 437 (code page 437) character set.

cp437 ASCII-to-EBCDIC translation table

Figure B-7: cp437 ASCII-to-EBCDIC translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0D	0E	0F
1_	10	11	12	13	B6	B5	32	26	18	19	1C	27	07	1D	1E	1F
2_	40	4F	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3_	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4_	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5_	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	4A	E0	5A	5F	6D
6_	79	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7_	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	C0	BB	D0	A1	3F
8_	68	DC	51	42	43	44	47	48	52	53	54	57	56	58	63	67
9_	71	9C	9E	CB	CC	CD	DB	DD	DF	EC	FC	B0	B1	B2	3E	B4
A_	45	55	CE	DE	49	69	9A	9B	AB	9F	BA	B8	B7	AA	8A	8B
B_	3C	3D	62	6A	64	65	66	20	21	22	70	23	72	73	74	BE
C_	76	77	78	80	24	15	8C	8D	8E	FF	06	17	28	29	9D	2A
D_	2B	2C	09	0A	AC	AD	AE	AF	1B	30	31	FA	1A	33	34	35
E_	36	59	08	38	BC	39	A0	BF	CA	3A	FE	3B	04	CF	DA	14
F_	EE	8F	46	75	FD	EB	E1	ED	90	EF	B3	FB	B9	EA	BD	41

cp437 EBCDIC-to-ASCII translation table

Figure B-8: cp437 EBCDIC-to-ASCII translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	EC	09	CA	1C	E2	D2	D3	0B	0C	0D	0E	0F
1_	10	11	12	13	EF	C5	08	CB	18	19	DC	D8	1A	1D	1E	1F
2_	B7	B8	B9	BB	C4	0A	17	1B	CC	CD	CF	D0	D1	05	06	07
3_	D9	DA	16	DD	DE	DF	E0	04	E3	E5	E9	EB	B0	B1	9E	7F
4_	20	FF	83	84	85	A0	F2	86	87	A4	5B	2E	3C	28	2B	21
5_	26	82	88	89	8A	A1	8C	8B	8D	E1	5D	24	2A	29	3B	5E
6_	2D	2F	B2	8E	B4	B5	B6	8F	80	A5	B3	2C	25	5F	3E	3F
7_	BA	90	BC	BD	BE	F3	C0	C1	C2	60	3A	23	40	27	3D	22
8_	C3	61	62	63	64	65	66	67	68	69	AE	AF	C6	C7	C8	F1
9_	F8	6A	6B	6C	6D	6E	6F	70	71	72	A6	A7	91	CE	92	A9
A_	E6	7E	73	74	75	76	77	78	79	7A	AD	A8	D4	D5	D6	D7
B_	9B	9C	9D	FA	9F	15	14	AC	AB	FC	AA	7C	E4	FE	BF	E7
C_	7B	41	42	43	44	45	46	47	48	49	E8	93	94	95	A2	ED
D_	7D	4A	4B	4C	4D	4E	4F	50	51	52	EE	96	81	97	A3	98
E_	5C	F6	53	54	55	56	57	58	59	5A	FD	F5	99	F7	F0	F9
F_	30	31	32	33	34	35	36	37	38	39	DB	FB	9A	F4	EA	C9

Default cp850 (code page 850) translation tables

This section contains the following tables:

- cp850 ASCII-to-EBCDIC translation table
- cp850 EBCDIC-to-ASCII translation table

The EBCDIC-to-ASCII translation tables in this section are the base tables for the predefined system cp 850 (code page 850) character set.

cp850 ASCII-to-EBCDIC translation table

Figure B-9: cp850 ASCII-to-EBCDIC translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0D	0E	0F
1_	10	11	12	13	3C	3D	32	26	18	19	1C	27	07	1D	1E	1F
2_	40	4F	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3_	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4_	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5_	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	4A	E0	5A	5F	6D
6_	79	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7_	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	C0	BB	D0	A1	3F
8_	68	DC	51	42	43	44	47	48	52	53	54	57	56	58	63	67
9_	71	9C	9E	CB	CC	CD	DB	DD	DF	EC	FC	70	B1	80	BF	FF
A_	45	55	CE	DE	49	69	9A	9B	AB	AF	BA	B8	B7	AA	8A	8B
B_	2B	2C	09	21	28	65	62	64	B4	38	31	34	33	B0	B2	24
C_	22	17	29	06	20	2A	46	66	1A	35	08	39	36	30	3A	9F
D_	8C	AC	72	73	74	0A	75	76	77	23	15	14	04	6A	78	3B
E_	EE	59	EB	ED	CF	EF	A0	8E	AE	FE	FB	FD	8D	AD	BC	BE
F_	CA	8F	1B	B9	B6	B5	E1	9D	90	BD	B3	DA	FA	EA	3E	41

cp850 EBCDIC-to-ASCII translation table

Figure B-10: cp850 EBCDIC-to-ASCII translation table

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	00	01	02	03	DC	09	C3	1C	CA	B2	D5	0B	0C	0D	0E	0F
1_	10	11	12	13	DB	DA	08	C1	18	19	C8	F2	1A	1D	1E	1F
2_	C4	B3	C0	D9	BF	0A	17	1B	B4	C2	C5	B0	B1	05	06	07
3_	CD	BA	16	BC	BB	C9	CC	04	B9	CB	CE	DF	14	15	FE	7F
4_	20	FF	83	84	85	A0	C6	86	87	A4	5B	2E	3C	28	2B	21
5_	26	82	88	89	8A	A1	8C	8B	8D	E1	5D	24	2A	29	3B	5E
6_	2D	2F	B6	8E	B7	B5	C7	8F	80	A5	DD	2C	25	5F	3E	3F
7_	9B	90	D2	D3	D4	D6	D7	D8	DE	60	3A	23	40	27	3D	22
8_	9D	61	62	63	64	65	66	67	68	69	AE	AF	D0	EC	E7	F1
9_	F8	6A	6B	6C	6D	6E	6F	70	71	72	A6	A7	91	F7	92	CF
A_	E6	7E	73	74	75	76	77	78	79	7A	AD	A8	D1	ED	E8	A9
B_	BD	9C	BE	FA	B8	F5	F4	AC	AB	F3	AA	7C	EE	F9	EF	9E
C_	7B	41	42	43	44	45	46	47	48	49	F0	93	94	95	A2	E4
D_	7D	4A	4B	4C	4D	4E	4F	50	51	52	FB	96	81	97	A3	98
E_	5C	F6	53	54	55	56	57	58	59	5A	FD	E2	99	E3	E0	E5
F_	30	31	32	33	34	35	36	37	38	39	FC	EA	9A	EB	E9	9F

APPENDIX C Troubleshooting

This appendix contains troubleshooting guidelines and process flow overviews for the Server Option. It contains the following topics:

For troubleshooting information about TRS, see the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*. For explanations of specific error messages, see the Mainframe Connect Client Option and Server Option *Messages and Codes*.

Where to start troubleshooting

At the client, the DirectConnect for z/OS Option workstation, and mainframe levels, check components systematically to locate the problem. Depending on your setup, you may want to check for problems in this sequence:

1. Connectivity
2. Client application
3. Client LAN
4. Client network setup
5. Major outage
6. DirectConnect for z/OS Option workstation
7. Connection from the client to the DirectConnect for z/OS Option workstation

8 Connection from the DirectConnect for z/OS Option workstation to the mainframe

9 Configuration between the transaction processor and the Server Option, or OmniSQL Access Module for DB2 for IMS TM and MVS

For any of these problems, the appropriate system administrator should use normal troubleshooting procedures. For example:

- Record specific information on the error message(s), including:
  - Error message number
  - Associated SNA sense codes or SNA Services error codes
  - Time the error occurred
  - The client or user affected
- Perform the recommended action.
- Continue the process until the problem is resolved.

Common problems and suggested solutions

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

This section contains the following subsections:

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

Configuration errors

This section contains the following sub-sections:

- Cannot establish session
- Session established, but the transaction does not run
- SDLC line or token-ring not up
- SDLC Link and PU are active, but the LU is not active

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

- For IMS TM, definitions for transaction name and APPLID
- SNA /NCP definitions for the LU and associated logmode
- SDLC or token-ring connection charts to the mainframe
- Sybase interface files for clients and TRS
- Sybase security definitions, including client logins, connection groups, and transaction groups

Verify that this information is the same as it was before the error occurred. If it is not, determine whether a recent change is contributing to the problem.

Cannot establish session

Cause

Any of the following:

- Mismatched LU definitions between SNA and workstation
- Mismatched modenames
- Incorrect SNA MODETAB and APPLID macros

Suggested action

Check the MVS system log on the mainframe for messages.

Correct the spelling.

Coordinate with the TRS administrator to check connection and modename profiles, using the utility shipped with the product.

Session established, but the transaction does not run

Cause

Any of the following:

- RACF security error
- Incorrect transaction ID in the TRS RPC table

Suggested action

Check the MVS system log on the mainframe for messages.

Verify definitions.

Coordinate with the TRS administrator for correct security and transaction ID setups.

SDLC line or token-ring not up

Cause

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

Suggested 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 for z/OS Option LU definition errors
- SSCPID value in the local LU profile set incorrectly

Suggested 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 IMS TM or SNA operator put the resource out of service with the vary command.
- 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 TRS administrator sees SNA Services timeout and connection failure messages when he or she tries to start the DirectConnect for z/OS Option, or
- The requesting client sees an SNA Services 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

This section explains what happens when line, adapter, or modem outages occur and how to help prevent them.

This section contains the following sub-sections:

- When these errors occur
- Preventing these errors

When these errors occur

Line, adapter, or modem outages result in error messages at the SNA console and at the DirectConnect for z/OS Option. The DirectConnect for z/OS Option records the message and, when possible, sends a similar error message to any affected clients.

Preventing these errors

Intermittent hardware errors and line degradation problems disrupt processing and may be difficult to find. It helps to check periodically for these problems. For example:

- To check for hardware errors, use the SNA error logs. 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 timeouts. Line degradation results in random SDLC line failures or very slow response to the client, even during a moderate processing load.

Troubleshooting at each component

This section explains troubleshooting at each of the Sybase components. It includes the following subsections:

- The Server Option support
- TRS support
- DirectConnect for z/OS Option Communications with the mainframe
- Gateway-Library support

The Server Option support

The Server Option support consists of several components on the IBM S/390 mainframe and the DirectConnect for z/OS Option platform, as the following diagram shows. These components provide tracing and logging, which you can use to locate errors.

TRS support

TRS does the following:

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

Each instance of TRS has a unique service name, which clients use to select a service for communication. Each instance of TRS has its own set of configuration information, defined globally in the Transaction Router Service Library (TRSL) configuration file.

As shown in the previous figures, TRS uses the following files:

- *<srvlibname>.tds* for tracing Sybase TDS traffic between TRS on UNIX platforms and mainframe SNA
- *ngtds.<srvlibname>* for tracing Sybase TDS traffic between TRS on PC platforms and mainframe SNA
- *svr.log* for logging TDS traffic between TRS and client workstations, and for recording errors

The Transaction Router Service Library (TRSL) SNATraceFile configuration property specifies the file to which the SNA side of the TRS trace file is written. Formatted TDS traces and errors are logged and written to this file.

The associated TRSL name is appended to these files so that the TRS administrator can differentiate the log for each instance of TRS. For more information, refer to the Mainframe Connect DirectConnect for z/OS Option *User's Guide for Transaction Router Services*.

The the DirectConnect for z/OS Option server logs TDS traffic between TRS and client workstations, and records errors. For more information, refer to the Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide for DirectConnect*.

DirectConnect for z/OS Option Communications with the mainframe

This section contains the following subsections:

- SNA LU 6.2
- IBM

TRS depends on the communications support of the server it runs on to communicate with the mainframe transaction processor. For IMS MVS, DirectConnect uses SNA LU 6.2.

SNA LU 6.2

The LAN communications server, such as SNA Services for AIX, uses the SNA trace file to record SDLC/SNA traffic between the workstation and mainframe. The vendor’s trace utility extracts this file.

For AIX and OS/2 platforms, the error log file records errors that SNA Services detects or Communications Manager detects. The IBM error log report utility extracts this information.

IBM

For IBM environments, debug or trace sockets are used. Refer to your IBM documentation for more information.

Gateway-Library support

Note Skip this section if you are using the DB2 UDB Option for CICS.

The Gateway-Library is a set of functions available for writing applications to enable mainframe environments to communicate with clients attached to TRS. These functions convert client calls into the TDS datastream needed to communicate with TRS and its clients.

Stubs provide access to the Gateway-Library functions. These stubs are a set of object libraries that application programmers can include in job steps used to link-edit programs they create.

Gateway-Library tracing functions enable you to trace program activity globally, for all transactions, or specifically, for individual transactions. Based on the transaction processor, tracing functions provide:

- API tracing for Gateway-Library calls, using the IMS TM System Log
- TDS header tracing, using the IMS TM System Log
- TDS data tracing, using the IMS TM System Log

Table C-1 shows the tracing functions:

Table C-1: Gateway-Library tracing functions

Function	Description
TDINFLOG	Determines what types of tracing are set
TDINFSPT	Indicates whether tracing is on or off for a transaction and returns the transaction ID
TDLSTSPT	Lists all transactions for which tracing is enabled
TDSETLOG	Turns system-wide tracing options on or off
TDSETSPT	Turns tracing on or off for a specific transaction
TDWRTLOG	Writes a user message or system entry

You can use standard IMS TM or MVS debugging tools or third party debugging tools to debug user-written applications.

For more information, see:

- Chapter 5, “Tracing and Accounting,” which describes the logging processes.
- The appropriate Mainframe Connect Server Option *Programmer’s Reference*, which describes Gateway-Library tracing functions. PL/1 and COBOL versions of this guide are available.

Remember that some TDS 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 are appended to this file until it is full; then the records are rejected.

Coordinating troubleshooting efforts

This section contains the following subsections:

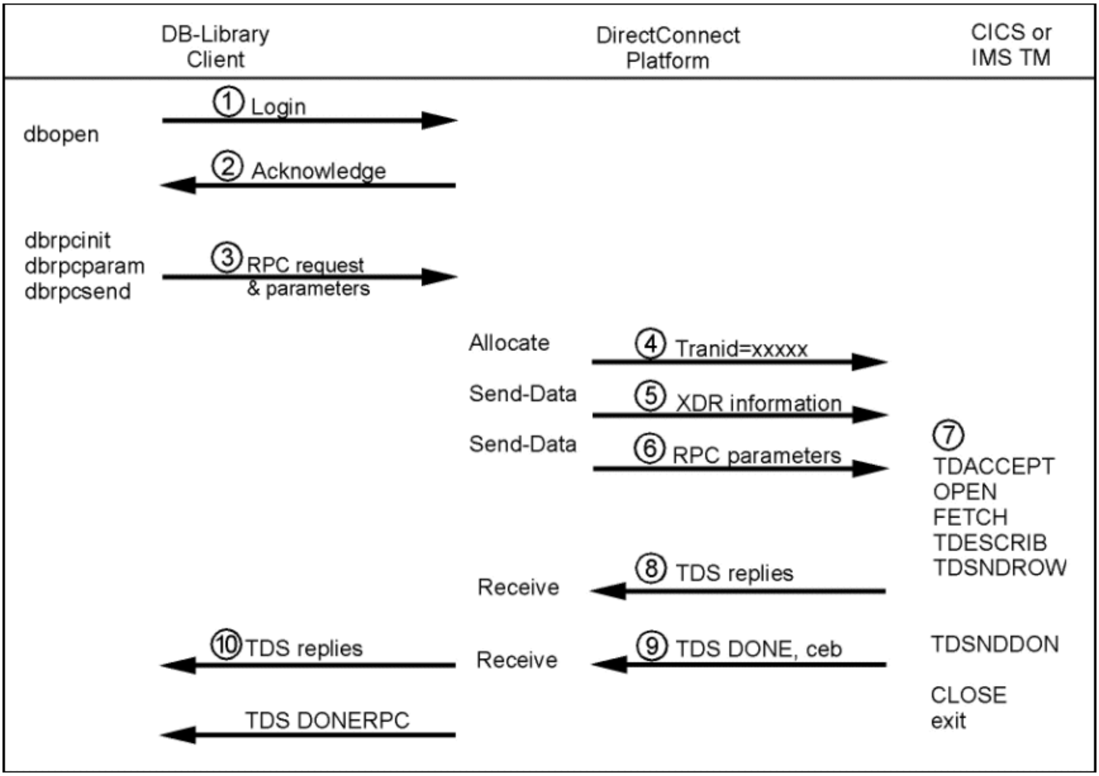
- Processing flow and requirements
- Process flow during attention sequences
- Browse applications

System administrators at the mainframe, TRS, and client need to coordinate troubleshooting efforts. To help you with your analysis, this section describes the processing flow from the client through TRS to the mainframe.

Processing flow and requirements

The following diagram shows the processing flow:

Figure C-1: Client-to-TRS-to-mainframe processing flow



The following steps describe the sequence shown above and highlight the requirements:

1 Assuming TRS started, the client opens a LAN connection to a designated DirectConnect for z/OS Option server and logs in. The following message may appear:

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

If so, make sure that:

- The client interfaces file is set up correctly.
- The client Sybase path variable (SYBASE) is defined correctly.
- The DirectConnect for z/OS Option server is specified in the DSQUERY variable.

2 On receiving the client login information, the DirectConnect for z/OS Option checks security as follows:

- If security is enabled, the DirectConnect for z/OS Option ensures that the client is authorized. If the client is not authorized, this error appears:

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

- If the client is authorized or security is disabled, the DirectConnect for z/OS Option 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 TRS over the logged-in LAN connection.

4 TRS receives the request and performs a table lookup to find the mainframe session and the Server Option transaction ID to use. The RPC and connection must be in the table. If security is enabled, the client must be authorized to use the RPC and connection to the mainframe. If the table lookup and security check are successful, the line is up, and the session is active, TRS allocates a conversation with the named transaction.

If a failure occurs during this process, SNA Services writes one of the following error messages to both the TRS log and the client:

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

The client is not authorized or is not listed correctly.

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

Connections to the mainframe are unavailable.

```
Request Rejected: Remote procedure ‘xxxx’ not
found.
```

The RPC name was entered incorrectly or the name is not in the lookup table.

5 TRS sends the client External Data Representation (XDR) information to the mainframe.

6 TRS sends the client RPC parameters to the mainframe, and then waits for a reply from the transaction.

7 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 XDR information and RPC parameters. The transaction also performs associated processing, such as issuing static SQL DB2 requests or reading VSAM or other database data.

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

9 TRS receives the 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, the LAN SNA software writes an error message to the DirectConnect for z/OS Option server log. It also writes an "Unexpected EOF from Adaptive Server Enterprise" error message to the client. (The mainframe is acting as a Adaptive Server Enterprise.) Gateway-Library tracing functions, if in use, also record errors in this process.

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

- The client may be set up to disconnect after invoking the transaction and before the transaction ends.
- Adaptive Server Enterprise logs out after sending a client request and, therefore, does not support long-running transactions.

For more information on identifying problems, see "Common problems and suggested solutions" on page 74.

Process flow during attention sequences

Any of the following actions results in an attention sequence:

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

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

1 Database-Library issues an attention packet to TRS, then discards anything else received until it receives a TDS DONE packet with the attention Ack bit on.

2 TRS converts the attention packet into a SNA SIGNAL command, issuing an LU 6.2 request-to-send verb. TRS then discards any results received from the mainframe until it receives a TDS DONE packet with the attention Ack bit on.

3 At the mainframe, IMS TM receives the SIGNAL and informs Server Option support.

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 appropriate Mainframe Connect Server Option *Programmer's Reference*. PL/1 and COBOL versions of this guide are available.

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.

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

1 Database-Library issues an attention packet to TRS, then discards anything else received until it receives a TDS DONE packet with the attention Ack bit on.

2 TRS converts the attention packet into a SNA SIGNAL command, issuing an LU 6.2 request-to-send verb. TRS then discards any results received from the mainframe until it receives a TDS DONE packet with the attention Ack bit on.

3. At the mainframe, IMS TM receives the SIGNAL and informs the Server Option support.
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 appropriate Mainframe Connect Server Option *Programmer's Reference*. PL/1 and COBOL versions of this guide are available.
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.

Browse applications

Sybase architecture uses a "streaming mode" of data transfer. Rather than sending a short block of data and waiting for a reply, the mainframe continuously sends data until the client stops accepting it. When the client stops accepting data, normal SNA pacing functions suspend data transfer.

For applications that select a small set of data, process it, then request the next block of data, it is best to use RPC parameters to specify the ID of a set of records. If the client RPC parameters are set up as return parameters, and the Gateway-Library TDSETPRM specifies the ID of the desired set of records, Server Option support returns the updated RPC value to the client. The client can use this value to invoke the next set of records.

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)**.

American Standard Code for Information Interchange

See **ASCII (American Standard Code for Information Interchange)**.

API

See **application program interface (API)**.

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 The Server Option for DB2 UDB as a client to The 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.

data source A collection of data, such as a database.

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.

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)**.

execute
To carry out an instruction.

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)**.

field The smallest identifiable part of a record.

file A collection of related data that is stored and retrieved by an assigned name.

format In programming languages, a language construct that specifies the representation, in character form, of data objects in a file.

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.

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 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.

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 The Server Option server. The server log file contains entries of events for each service managed by the 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)**.

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 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 The 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.

platform The operating system environment in which a program runs.

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 The 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 The 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.

standalone 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**.

syntax

The rules for how to construct a statement.

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 The 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.

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.

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