

SYBASE®

Installation and Administration Guide

Mainframe Connect™ Client Option

15.0

[IBM CICS]

DOCUMENT ID: DC34012-01-1500-01

LAST REVISED: August 2007

Copyright © 1991-2007 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

About This Book	vii
CHAPTER 1 Understanding the Client Option	1
What is the Client Option?	1
Adaptive Server Enterprise (ASE)	2
Server Option	2
Architecture	2
Choosing a network driver	2
General criteria for choosing a driver	3
Client Option communications.....	3
Client Option environment	4
Communication flow	6
Understanding TCP/IP SYGWXCPH definitions	7
Client Option security	7
CHAPTER 2 Planning Your Installation	9
Installation media	9
Pre-installation tasks	9
1. Verify the platforms, components, and distributed software	10
2. Determine JCL and system information	10
3. Determine CICS and DB2 UDB information	10
4. Determine compiler information	11
5. Determine Client Option information	11
6. Determine FTP information	12
7. Plan the security requirements	12
8. Identify the change control requirements	12
9. Back up the release libraries (upgrades only)	12
10. Determine the library names	13
11. Verify the connectivity	13
CHAPTER 3 Installation and Configuration	15
Installation and configuration	15
Libraries and samples	21

CHAPTER 4	Using isql.....	23
	Introducing isql	23
	Invoking isql and initializing the environment	24
	Establishing the connection	25
	Entering SQL commands	26
	Viewing data results	27
	Viewing trace results	27
APPENDIX A	Customization Options	29
	Overview	29
	Customizing global options (SYGWM CST).....	30
	Using the IBM z/OS conversion environment and services	33
	Customizing mainframe character set conversion options	
	(SYGWM CXL)	33
	Overriding the supplied SBCS translation tables	34
	Defining new SBCS translation tables.....	35
	Defining new character set entries.....	38
	Customizing dynamic network drivers (SYGWDRIV).....	39
	CICS network drivers	39
	Customizing the TCP/IP driver (SYGWHOST)	41
	Macro formats	41
	Macro parameters	41
	Defining license keys (SYGWLKEY).....	42
	Building a global customization module (SYGWXCPH).....	43
APPENDIX B	Translation Tables	45
	Understanding the ASCII-EBCDIC and EBCDIC-ASCII translation	
	tables	45
	Default ASCII_8 translation tables	47
	ASCII_8, ASCII-to-EBCDIC translation table	48
	ASCII_8, EBCDIC-to-ASCII translation table	48
	Default ISO_1 translation tables	50
	ISO_1 ASCII-to-EBCDIC translation table	51
	ISO_1 EBCDIC-to-ASCII translation table	51
	Default cp437 (code page 437) translation tables.....	53
	cp437 ASCII-to-EBCDIC translation table.....	54
	cp437 EBCDIC-to-ASCII translation table.....	55
	Default cp850 (code page 850) translation tables.....	55
	cp850 ASCII-to-EBCDIC translation table.....	57
	cp850 EBCDIC-to-ASCII translation table.....	58
APPENDIX C	Setting Up the CICS Sockets Interface	59

Understanding the CICS sockets interface 59
 Installing and configuring the Client Option of CICS sockets interface
 60

APPENDIX D Setting Up Secure Sockets Layer Protocol 61
 Understanding Secure Sockets Layer..... 61
 Description of SSL features..... 61
 How SSL provides security 62
 Setting up SSL in Sybase products..... 63
 Setting up SSL in ASE and Open Client 63
 Setting up SSL in IBM z/OS 65
 Using System SSL on z/OS 66
 Configuring AT-TLS..... 67
 AT-TLS support in Client and Server Options for CICS 70
 Configuring a z/OS client or server system 71
 For more information 78

Glossary 79

Index 95

About This Book

The Mainframe Connect™ Client Option for CICS *Installation and Administration Guide* describes how to install and configure the Client Option for CICS. It includes planning considerations, installation instructions, configuration instructions, and troubleshooting information.

Note If you want to go directly to the installation planning tasks, 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/OS mainframe computer. This book refers to anyone performing these tasks as the Client Option administrator.

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

How to use this book

This table shows where to find the information you need in this book:

To	See
<i>Understand</i> the Client Option	Chapter 1, “Understanding the Client Option”
<i>Plan</i> for installation and configuration	Chapter 2, “Planning Your Installation.”
<i>Install</i> the Client Option	Chapter 3, “Installation and Configuration”
<i>Access</i> a server or host-based database using the Interactive SQL utility (isql)	Chapter 4, “Using isql”
<i>Customize</i> the Client Option	Appendix A, “Customization Options”
<i>Reference</i> default translation tables to customize SBCSS	Appendix B, “Translation Tables”
<i>Install and configure</i> the CICS sockets interface	Appendix C, “Setting Up the CICS Sockets Interface”
<i>Install and configure</i> the Secure Sockets Layer (SSL) protocol	Appendix D, “Setting Up Secure Sockets Layer Protocol”

Related documents

You may also need to refer to this related documentation:

- Mainframe Connect Client Option and Server Option *Messages and Codes*
- Mainframe Connect Client Option *Programmers Reference for C*
- Mainframe Connect Client Option *Programmers Reference for COBOL*
- Mainframe Connect Client Option *Programmers Reference for PL/I*
- Mainframe Connect Client Option *Programmers Reference for Client Services Applications*

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 Client Option uses 8-character function names, while other versions of Client-Library™ use longer names. This book uses the long version of Client-Library names with one exception: the 8-character version is used in syntax statements. For example, CTBCMDPROPS has 11 letters. In the syntax statement, it is written CTBCMDPR, using 8 characters. You can use either version in your code.

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

Symbol	
()	Parentheses indicate that parentheses are included as part of the command.
{ }	Braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you type the option.
[]	Brackets indicate that you can choose 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.

The following table explains style conventions used in this book.

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
Directory names, path names, and file names	<i>/usr/bin directory, interfaces file</i>
Variables	<i>n bytes</i>
Adaptive Server® Enterprise datatypes	datetime, float
Sample code	01 BUFFER PIC S9(9) COMP SYNC. 01 BUFFER PIC X(n) .
User input	01 BUFFER PIC X(n)
Client-Library and Gateway-Library function argument names	<i>BUFFER, RETCODE</i>

This type of information	Looks like this
Client-Library function arguments that are input (I) or output (O)	<i>COMMAND</i> – (I) <i>RETCODE</i> – (O)
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 appear in this typeface.

Accessibility features

The HTML documentation has been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at <http://www.sybase.com/accessibility>. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

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.



Understanding the Client Option

Topic	Page
What is the Client Option?	1
Architecture	2
Choosing a network driver	2
Client Option communications	3

What is the Client Option?

The Client Option is a programming environment that provides Open Client™ Client-Library routines you can use to build mainframe client applications.

The Client Option runs on an IBM z/Series or plug-compatible mainframe computer. It uses the LU 6.2 or TCP/IP communications protocols and is available for CICS.

Note The Client Option is also available for IMS TM and native MVS. This document pertains to the Client Option for CICS only.

Client Option applications can communicate with two kinds of servers:

- Adaptive Server® Enterprise (ASE) and Open Server™ on PCs and several mid-range UNIX platforms
- Server Option applications running in a CICS region on the mainframe

Client Option applications can send requests to ASE and Server Option applications.

Adaptive Server Enterprise (ASE)

Client Option applications can send requests to an ASE in a two-tier (gateway-less) environment using TCP.

Server Option

Client Option applications can send requests directly to the Server Option running in a CICS region. With TCP/IP, the Client Option can send requests to the Server Option running in the same CICS region.

Architecture

The Client Option works in a two-tier, gateway-less environment.

A Client Option network configuration consists of:

- A host-based client, which is a Client Option program running under CICS. The client program selects a server and sends requests to that server.
- A server, which can be any server that Client Option applications can access, including servers on the LAN—Sybase Open Servers and Adaptive Servers—as well as Server Option running in a separate CICS region. If using TCP/IP, the Client Option may access servers running in the same CICS region.

For more information on using Client-Library functions, see the appropriate Mainframe Connect Client Option *Programmers Reference*. COBOL, PL/1, and C versions of this reference guide are available.

Choosing a network driver

The Client Option supports concurrent use of multiple network drivers that can be invoked from the same Client Options and Server Options common code base. The appropriate network driver is loaded dynamically at the time the program executes.

You must use the SYGWDRIV macro to define the network drivers to be used with the Client Option and the Server Option. The default SYGWXCPH member provided contains the SYGWDRIV macro definitions for *all* the supported network drivers. When installing the Client Option, be sure to edit the appropriate IxHOST member to comment-out the drivers that your site does not intend to use.

This section provides an overview of network communication, and lists and describes general criteria for choosing a driver.

General criteria for choosing a driver

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

CICS environment

These drivers are supported in the CICS environment:

- TCP/IP for an IBM network
- LU 6.2 for an SNA network

The following table indicates which drivers can be used by the Client Option for CICS in a two-tier environment to communicate with the Server Option for CICS or for IMS and MVS.

Driver	Gateway-less	To Server Option for CICS	To Server Option for IMS and MVS
<i>LU62CICS</i>		X ¹	X
<i>TCPCICS</i>	X		

1. The Client Option for CICS and the Server Option for CICS run in separate CICS regions.

Client Option communications

This section describes Client Option communications. It also explains the communication flow.

Client Option environment

The Client Option routes requests to servers on the LAN or on the mainframe. When using a TCP/IP network driver, Client Option requires the following additional information to establish a TCP/IP connection to a given server:

- The name of the IBM TCP/IP address space
- IP address or host name of the host on which the server runs
- Server name

When an application requests a connection to a server, Client Option:

- Looks up the server name in the TCP/IP Server-Host Mapping Table (SYGWHOST)
- Determines the type of listener the server is using. If the listener is for a Sybase LAN or mainframe, the Client Option routes the request directly to the server

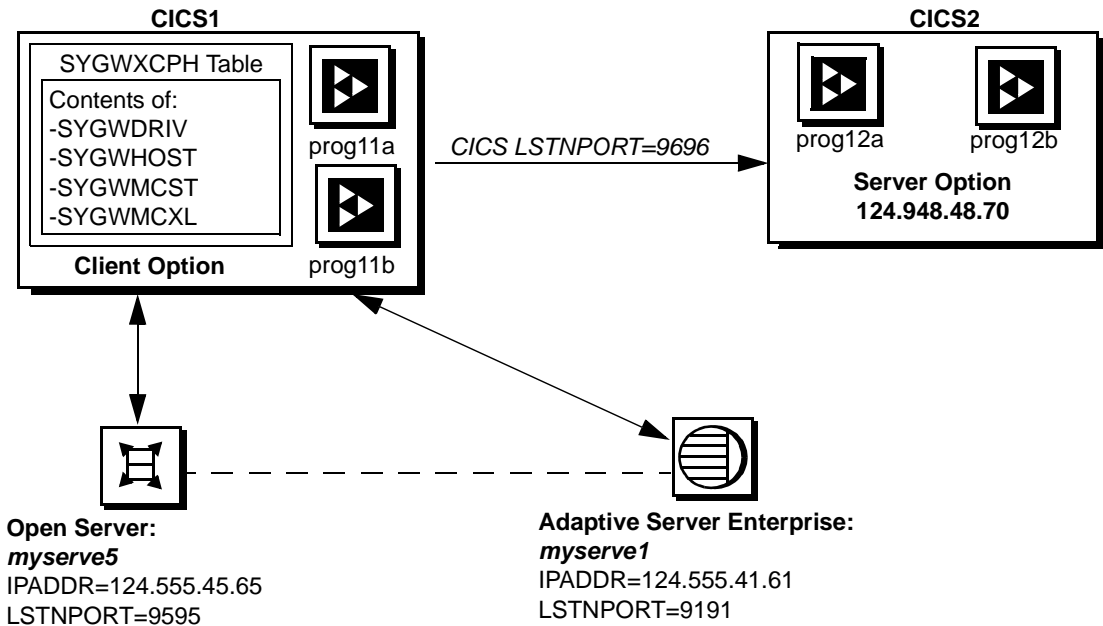
After the server has received and processed the request, Client Option reads the results directly from a LAN-based server or mainframe Open Server.

Sample network

Figure 1-1 provides a sample two-tier TCP/IP environment. The Client Option is running in CICS region *CICS1*, and the Server Option is running in *CICS2*.

Note Using TCP/IP, the Client Option has the capability to connect to an Open Server running in the same CICS region.

Figure 1-1: Sample two-tier/gateway-less Client Option network

**SYGWHOST entries:**

```
SYGWHOST TYPE=ENTRY
  IBMTCPADDRSPACE=TCPIP,
  IPADDR=124.555.45.65,
  LSTNPORT=9595,
  LISTENER=LAN,
  SERVERNAME=MYSERVE5
```

```
SYGWHOST TYPE=ENTRY
  IBMTCPADDRSPACE=TCPIP,
  IPADDR=124.555.41.61,
  LSTNPORT=9191,
  LISTENER=LAN,
  SERVERNAME=MYSERVE1
```

In Figure 1-1, CICS1 has connections to:

- CICS2, the region where Server Option is installed
- A local area network (LAN) that has an ASE (myserve1) and an Open Server (myserve5)

Client Option programs can send requests to Server Option and to any LAN server. Figure 1-1 shows Client Option prog11a and prog11b sending requests to Server Option prog12a and prog12b.

Communication flow

This section describes what happens at the mainframe and at the server during Client Option processing.

At the mainframe

A Client Option application calls a pre-written procedure, such as a stored procedure or a Server Option application. All calls from the Client Option to remote nodes are processed using the TCP/IP communications protocol. For requests to an Open Server (LAN), the client can access any data available to the Open Server application. If the request is to the Server Option, the client can access any data storage system accessible through CICS.

The called procedure or transaction executes and returns results to the calling Client Option application, which can use the results for local processing. If the client has permission, the client transaction can update data at remote sites by inserting, modifying, and deleting entries in database tables or other data storage systems.

isql utility

The Client Option includes isql, a utility that allows users to send SQL language commands interactively. Users specify the server and whether to enable tracing, and then type SQL commands in a 3270 panel. For more information, see Chapter 4, “Using isql.”

Server-Host Mapping Table

For TCP/IP, the Client Option includes a Server-Host Mapping Table that allows you to define servers. For more information, see “Customizing the TCP/IP driver (SYGWHOST),” in Appendix A, “Customization Options.”

At the server

Typically, a server accepts requests from a client and returns results. In a Client Option environment, the server can be an ASE, an Open Server on the LAN, or a Server Option on the mainframe.

From the server standpoint, a request from an IBM host is no different than a request from a Sybase client. The Client Option participates in ASCII-EBCDIC and EBCDIC-ASCII translations and datatype conversions.

Understanding TCP/IP SYGWXCPH definitions

If you are using a TCP/IP driver, you must use the SYGWHOST macro to define the TCP/IP Server-Host Mapping Table.

For a description of the macro parameters and formats, see Appendix A, “Customization Options.”

Client Option security

Security for Client Option processing can be configured to require permission to:

- Log in to the target server or desired CICS region
- Use specific commands, stored procedures or transactions, and data objects at the target server

For more information about:

- *ASE security*: Refer to the chapter called “Security Administration,” in the *Adaptive Server Enterprise System Administration Guide*.
- *Mainframe security*: Refer to documentation provided with CICS and z/OS, or with the appropriate mainframe security system.
- *Secure Sockets Layer (SSL) implementation and configuration*: Refer to Appendix D in this guide.

Topic	Page
Installation media	9
Pre-installation tasks	9

Installation media

The Client Option is distributed on CD or in downloadable form.

Note The Client Option is distributed on CD or in downloadable form. For information on obtaining the latest EBFs for the Client Option, see the Mainframe Connect Client Option for CICS *Release Bulletin*.

Pre-installation tasks

Installation requires completing the following pre-installation tasks, which are explained in the following subsections.

Note Skip any steps that do not pertain to the options you have chosen to install.

1. Verify the platforms, components, and distributed software
2. Determine JCL and system information
3. Determine CICS and DB2 UDB information
4. Determine compiler information
5. Determine Client Option information

6. Determine FTP information
7. Plan the security requirements
8. Identify the change control requirements
9. Back up the release libraries (upgrades only)
10. Determine the library names
11. Verify the connectivity

1. Verify the platforms, components, and distributed software

See the Mainframe Connect Client Option for CICS *Release Bulletin*.

2. Determine JCL and system information

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

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

3. Determine CICS and DB2 UDB information

Determine the following information 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 (DSN)* – name of the CICS RDO (DFHCSD) containing the application resource definitions used by your CICS region.

- *RDO group list* – RDO group list used by your CICS region when executing an initial start.
- The *CICS region APPLID* – VTAM APPLID for your CICS region.
- *DB2 system data sets high-level qualifier* – high-level qualifier used for DB2 system data sets.
- *DB2 exit data set name*.
- *DB2 data set name (DSN)*.

4. 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* – module used to execute COBOL in your environment.
- *COBOL compiler LOADLIB* – system LOADLIB where your COBOL compiler module resides.
- *PL/1 compiler name* – module used to execute PL/1 in your environment.
- *PL/1 compiler LOADLIB* – system LOADLIB where your PL/1 compiler module resides.
- *C compiler data sets high-level qualifier* – high-level qualifier used for C.
- *TCP/IP data sets high-level qualifier*.

5. Determine Client Option information

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

- *TCP address space name*.
- *Remote server name* – name by which your Client Option applications will refer to the remote server.
- *Remote TCP host name* – DSN name for the remote server.
- *Remote server TCP host port* – TCP/IP port used by the remote server.

6. Determine FTP information

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

- *User ID.*
- *Password.*
- *Mainframe host name.*
- *Control port number* – listener port used by your mainframe FTP server, usually 21.
- *TCP/IP address space name.*
- *Volume serial number or unit* – specifies either a volume serial number (VOL=SER) and unit assignment for FTP to use, or allows FTP to use default values.
- *Log path name* – indicates where FTP log information will be written.

7. Plan the security requirements

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

8. Identify the change control requirements

Create a change control plan that includes:

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

9. Back up the release libraries (upgrades only)

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

10. Determine the library names

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

You do not need to remove previous releases from your Sybase libraries because default names shipped with this release create an entirely unique set of release libraries. However, you can change them, 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 regions, 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.

11. Verify the connectivity

Use the standard LAN ping utility to ensure connectivity between z/OS and the workstation running ASE.

Topic	Page
Installation and configuration	15
Libraries and samples	21

Installation and configuration

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

Licensing information

The Client Option for CICS requires a permanent authorization key. However, Sybase includes a temporary key, which is valid for 30 days, within the order at installation time. To avoid interrupting your operations, call Customer Service at 1-800-8Sybase (1-800-879-2273), select Option 3, then select Option 3 again, and request a permanent key.

When speaking with Customer Service, be sure to have the following information ready:

- Product name
- Order number
- For the machine you are using:
 - Serial number
 - Machine type
 - Model number
- A valid e-mail address

Note Please allow seven business days for the key to be generated and sent to you.

The two procedures in this section describe the installation steps you need to install all Mainframe Connect options from the InstallShield wizard and to complete the installation for the Client Option for CICS. You can skip those installation steps that do not pertain to the options you have chosen to install.

Note The InstallShield wizard runs only on Windows.

❖ **To install from the InstallShield wizard**

- 1 Start the InstallShield wizard from CD by executing *setupwin.exe*, which is in the root directory.
- 2 The dialog box displays the options available for installation. To continue the installation, click Next.
- 3 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. Click Next.
- 4 Select the components you want to install.

Note If you are installing the Server Option for CICS or the DB2 UDB Option for CICS, the Server Option for CICS runtime component is automatically selected as you proceed to the next window.

Click Next.

- 5 Provide the following JCL and system information:
 - *JCL Line 1-3* – (a valid jobcard) used to run the final installation jobs in TSO.
 - *High Level Qualifier* – used as a prefix for all data sets generated during installation.
 - *Volume* – indicates where generated data sets are cataloged.
 - *Unit* – (unit parameter value) indicates the device requirements for cataloging generated data sets.
 - *Work Unit* – for the use of temporary work data sets.
 - *Customer CICS, IMS, and MVS Loadlibs* – 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 are installing an option that uses CICS, DB2 UDB, or IMS, provide this information where it applies. Otherwise, skip to the next step.
 - *CICS system datasets hlq* – high-level qualifier used to locate *SDFHLOAD* and other CICS libraries.
 - *RDO Dataset* – the name of the CICS RDO (DFHCSD) containing the application resource definitions used by your CICS region.
 - *RDO Group List* – the RDO group list used by your CICS region when executing an initial start.
 - *CICS Region Applid* – the VTAM APPLID for your CICS region.
 - *DB2 system datasets hlq* – high-level qualifier used for DB2 system data sets.
 - *DB2 Exit Dataset* – the name of the DB2 exit data set used by your DB2 region.
 - *DB2 DSN Name* – the data set name (DSN) of your DB2 region.
 - *IMS datasets hlq* – high level qualifier used to locate IMS libraries.

Click Next.

- 7 If you are installing 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* – high-level qualifier used for the Language Environment 370 that is used here to locate data sets like *CEELKED*.
 - *COBOL Compiler Name* – the module used to execute COBOL in your environment.
 - *COBOL Compiler Loadlib* – the system LOADLIB in which your COBOL compiler module resides.
 - *PLI Compiler Name* – the module used to execute PLI in your environment.
 - *PLI Compiler Loadlib* – the system LOADLIB in which your PLI compiler module resides.
 - *C compiler datasets hlq* – high-level qualifier used for C to locate data sets like *SBCOMP*.

- *TCP/IP datasets hlq* – high-level qualifier used to locate data sets like *SEZATCP*.

Click Next.

- 8 If you are installing 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* – the name of your TCP/IP region.
- *Server Name* – the name by which your Client Option applications refers to the remote server.
- *Server TCP Host Name* – the DNS name for the remote server.
- *Server TCP Host Port* – the TCP/IP port used by the remote server.

Click Next.

- 9 If you are installing 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* – name of your TCP/IP region.
- *Listener Port* – 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* – mainframe user ID for the FTP session.
- *Password* – password for the FTP SESSION.
- *Mainframe Host Name* – mainframe DSN name.
- *FTP Port* – 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* – indicates where FTP log information is to be written. This log information may be useful in troubleshooting FTP PROBLEMS.
- 12 Click Next. The InstallShield wizard will create JCL and upload the selected components to your mainframe.
 - 13 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 that 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

❖ **To complete the installation**

- 1 Locate the installation JCL for the Client Option for CICS in *hlq.OCC150.CICS.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* – runs IKJEFT01 to use the TSO RECEIVE command to build and populate the product libraries.
 - *IxRDO* – runs the CICS Resource Definition Utility, DFHCSDUP, to define the transaction, program, and file entries for the Client Option for CICS. If your CICS region has had a previous version of the Client Option, you may need to un-comment or change the DELETE and REMOVE entries at the top of the RDO input.

- *IxVSAM* – allocates the VSAM data sets used for error and trace logging. Because these data sets are shared by the Client Option for CICS, the Server Option for CICS, and the DB2 UDB Option for CICS, they may already have been created.
- *IxHOST* – assembles and links the Client Option for CICS customization module, character sets, remote host definitions, and license keys. You may rerun this job at any time to change configuration and character sets or to add, remove, or modify remote host definitions and license keys.

Note The Server Option for CICS installation has its own *IxHOST* job. If you are installing both the Client and Server Options for CICS, you should run the *IxHOST* job contained in *hlq.OSC150.CICS.JCL*.

- *IxDELETE* – (optional) deletes the data sets in the TSO XMIT form used for the installation.
- 3 Run these jobs if you want to compile and link-edit the sample applications provided with the Client Option for CICS:
- *SCTCOB1* – compiles and links the sample COBOL applications that use the Client Option interface.
 - *SCTCOB2* – compiles and links the sample COBOL applications that act as Server Option for CICS applications that make Client Option calls.

Note This job can be run only if the Server Option for CICS has been installed.

- *SCTC1* – compiles and links the sample C applications that use the Client Option interface.
- *SCTPLI1* – compiles and links the sample PL/1 applications that use the Client Option interface.
- *SCTPLI2* – compiles and links the sample PL/1 applications that act as Server Option for CICS applications that make Client Option calls.

Note This job can be run only if the Server Option for CICS has been installed.

- *SCSACOB* – compiles and links the sample COBOL applications that use the CSA interface.

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.

Topic	Page
Introducing isql	23
Invoking isql and initializing the environment	24
Establishing the connection	25
Entering SQL commands	26

Introducing isql

Using the Interactive SQL (isql) utility, you can type requests directly onto a 3270-type window. The Client Option routes SQL requests to the requested server and displays the results in a window at your terminal. No programming is required.

From the mainframe, you can also use isql to access ASE, Open Server, or host-based databases through the Server Option. The server is specified in the login window.

Function and action keys
for isql

This chapter describes the program windows and the entry choices available on them. Table 4-1 shows the keyboard commands for isql on these windows:

Table 4-1: isql function and action keys

Key	Action
Clear	Ends the current transaction and exits from the current window to the CICS command window.
Enter	Performs the indicated action.
Tab	Moves the cursor to the next field.
PF1	Does one of the following: <ul style="list-style-type: none">• Resets the fields to blank.• Resets the temporary storage browse pointer to the beginning of the queue.
PF3	Does one of the following: <ul style="list-style-type: none">• Terminates the connection to the server or, if not connected, terminates the environment.• For viewing isql results, terminates the current result set. If you are viewing the last or only result set, it returns to the SQL input display.
PF4	Terminates all result sets and returns to the SQL input display.
PF7	Scrolls backward.
PF8	Scrolls forward.
PF9	Displays messages.
PF10	Shifts left.
PF11	Shifts right.

Invoking isql and initializing the environment

❖ To invoke isql and initialize the environment

- 1 Enter the isql command on a 3270 terminal window to invoke the Open Client Environment window.

When you open this window, it invokes isql.

- 2 To set tracing, enter:

- Y – to enable tracing.

During the current isql session, each time you press Enter after keying a value in a window or typing a SQL command, the Trace/Error window displays trace information results.

- N (or blank) – to proceed without tracing.

When you press Enter after typing a value or SQL command, the SQL Result Display window displays results.

3 To set whether to save results for browsing, enter:

- Y – to save results to temporary storage.

This allows you to scroll backward as well as forward through returned rows. Otherwise, you can scroll forward only.

- N (or blank) – to proceed without saving results to temporary storage.

Whether or not you save results, you can scroll left and right, if needed, to view the entire length of returned rows.

Refer to Table 4-1 for isql function and action keys.

Establishing the connection

Use the Establish Open Client Server Connection window to establish a connection with a specific server. The server can be an ASE, an Open Server application, or a separate CICS region on either the same or a different mainframe.

❖ To establish a connection to the server

1 At the initialization window, press Enter to display the Establish Open Client Server Connection window.

Note At the bottom right of the window, the Temp Stg field indicates whether or not you chose to scroll backwards, that is, not to save results to temporary storage. (You set this parameter on the previous window.)

2 To connect to a specific server, enter the requested values:

- Server name – name of the server you want to access. This name must already be defined in the Connection Router Table.
- Login name – your login ID for the specified server. This login ID is passed to the server.
- Password – password you use to log in to the specified server. This password is passed to the server.
- Net-Driver – type of network driver you use (IBMTCP/IP or LU62).

- Login Charset – character set with which to log in to the remote server. Default value are:
 - If the USEIBMUNICODE parameter in *SYGWM CST* is set to Y, the default character set is utf8.
 - If the USEIBMUNICODE parameter in *SYGWM CST* is set to N, the default character set is iso_1.
 - Display CCSID – CCSID to which query result sets should be converted.
- 3 When you press Enter, the isql utility establishes a connection with the named server and displays the Enter Open Client ISQL Commands window.

Entering SQL commands

❖ **To enter SQL commands**

- 1 Go to the Enter Open Client ISQL Commands window to submit a SQL command to the specified server. The accessed DBMS must recognize these SQL commands.
- 2 To send SQL commands to the server, enter a single SQL statement at the prompt, for example:

```
select * from SYBASE.SAMPLETB
```

Note To get multiple results, first type “begin,” then type several lines without a joining semicolon, and then type “end,” for example:

```
begin
select firstname from sampletb
select lastname from sampletb
end
```

When you press Enter, the server processes the SQL command and returns the results. If you turned tracing on or specified that results be saved to temporary storage, the server saves the results to temporary storage.

Viewing data results

You can view the results of your SQL command on the SQL Result Display window as follows:

- If tracing is *not* turned on, the SQL Result Display window appears after you submit your SQL command.
- If tracing is turned on, the Trace/Error Display window displays the results of your SQL command. To display the SQL Result Display window, press Enter.

If result tables are larger than your screen, use the PF keys indicated on the display to scroll through the results.

Note You can scroll backward (PF7) *only* if when you initialized, you chose to save results to temporary storage.

Viewing trace results

The Trace/Error Display window displays after you finish typing a window. Error messages display at the top. For more information about the error messages, see the Mainframe Connect Client Option and Server Option *Messages and Codes* and the Mainframe Connect Client Option *Programmers Reference*. COBOL, C, and PL/1 versions of this guide are available.

Using the previous SQL command from step 2 in “Entering SQL commands” on page 26, the following results are shown on the Trace/Error Display window when underlying Client Option Client-Library routines are executed:

```
ct_con_alloc_rc -1
cd_diag (CS_INIT) rc+-1
ct_con_props (CS_USERNAME) rc=-1
ct_con_props (CS_PASSWORD) rc=-1
ct_connect rc=-1
```

For more information about individual Client-Library routines, refer to the Mainframe Connect Client Option *Programmers Reference*. COBOL, C, and PL/1 versions of this guide are available.

Customization Options

Topic	Page
Overview	29
Customizing global options (SYGWCST)	30
Using the IBM z/OS conversion environment and services	33
Customizing mainframe character set conversion options (SYGWCXL)	33
Customizing dynamic network drivers (SYGWDRIV)	39
Customizing the TCP/IP driver (SYGWHOST)	41
Defining license keys (SYGWLKEY)	42
Building a global customization module (SYGWXCPH)	43

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:

- SYGWCST – a global customization macro.
- SYGWCXL – a character set conversion macro..
- SYGWDRIV – specifies which dynamic network drivers are used at the site.
- SYGWHOST – provides mapping between Sybase Server names and TCP/IP addresses or host names.
- SYGWLKEY – 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.

Table A-1 describes SYGWMCST parameters. Except where noted, these apply to both the Client Option for CICS and Server Option for CICS.

Table A-1: Complete list of SYGWMCST parameters

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

Parameter	Default	Format	Purpose
DECPOINT (Server Option only)	'.'	Either a decimal point or comma delimited by single quotation marks	Decimal point indicator, used 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. <i>For CICS only:</i> 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 <i>Users Guide for Transaction Router Services</i> .

Parameter	Default	Format	Purpose
MVSDDNAME <i>(IMS TM and MVS only)</i>	blank	From 1 to 8 characters	<p>The DD name of the MVS Open Client and 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.</p> <p>MVSDDNAME must match a DD name specified in the JCL for one of the following:</p> <ul style="list-style-type: none"> • 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	<p>Designates the default national language used by the Client Option or Server Option. Also see the CHARSETSRV property.</p>
ROWLIMIT <i>(Server Option only)</i>	0 (zero)		<p>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.</p> <p>ROWLIMIT=<i>n</i>, where <i>n</i> indicates the global limit of rows that can be sent.</p>
USEIBMUNICODE	N	Y or N	<p>Specifies whether or not Unicode support for a particular z/OS installation is enabled through the IBM conversion environment and services.</p> <ul style="list-style-type: none"> • If USEIBMUNICODE=Y, IBM support is used for character set conversions. • If USEIBMUNICODE=N, conversion is accomplished through the product-supplied translation tables. <hr/> <p>Note If USEIBMUNICODE=Y, all character sets that are to be used at a particular site must have entries created with the SYGWMCXL macro.</p> <hr/>

See “Using the IBM z/OS conversion environment and services” on page 33.

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. When the conversion environment and services are 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

- 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 this command:

```
SET UNI=01
```

This 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 SYGWMCSST macro in the SYGWXCPH customization module. For more information on installing Unicode support for IBM z/OS, see “Support for Unicode Using Conversion Services” (SA22-7649-07).

Customizing mainframe character set conversion options (SYGWMCXL)

SYGWMCXL is the character set conversion macro in the SYGWXCPH table. 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

Warning! 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. Table A-2 shows the SBCS settings of the parameters for SYGWMCXL:

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: <ul style="list-style-type: none"> • 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 for the Server Option in a three-tier environment, you must 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 43.

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`

Warning! 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.

Overriding ASCII-to-EBCDIC 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 Figure A-1, the client is using `us_english`, which is not predefined.

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-40,0A-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 next 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.
CHARSETTYPE	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.

Note If USEIBMUNICODE=Y, all character sets that are to be used at a particular site must have entries created with the SYGWMCXL macro.

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, these 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 the Server Option.

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

CICS network drivers

Table A-3 shows the default drivers that 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 these macro definitions, which set up support for all three 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

```

Using the 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 window

```

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 8-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 host names. Do not depend on the default shipped with the installation to work in your environment.

Macro formats

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

Note For the Server Option, only the TYPE=INITIAL and TYPE=FINAL macros are required. For the Client Option, only the TYPE=ENTRY macro is required.

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
```

Macro parameters

There are six parameters in the SYGWHOST 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: <ul style="list-style-type: none"> • 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 seven characters preceded by “&&”. For example, the symbolic name “SYBTCP” would be designated as follows: <pre>IBMTCPADDRSPACE=&&SYBTCP</pre> 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.
LISTENER	One of the following: <ul style="list-style-type: none"> • <i>LAN</i> if the listen port is for a LAN-based server (default) • <i>CICS</i> if the listen port is for an CICS Server Option listener • <i>IMS</i> 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.

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 runtime.

There are two parameters in this macro:

Parameter	Definition
PRODUCT	The product related to the license key, either the Client Option, the Server Option, or the DB2 UDB Option. Valid values are OCC, OSC, or DB2.
KEY	Defines the license key given for a product. The license key is a 23-character numeric value.

This example of SYGWLKEY defines license keys for four Mainframe Connect options in the order they are listed: Client Option for CICS, Server Option for CICS, Server Option for IMS and MVS, and DB2 UDB Option for CICS:

```
SYGWLKEY TYPE=INITIAL
SYGWLKEY TYPE=ENTRY, PRODUCT=OCC, KEY=19320-00000-10$*#-#19$B
SYGWLKEY TYPE=ENTRY, PRODUCT=OSC, KEY=19300-00000-10E2G-4K##6
SYGWLKEY TYPE=ENTRY, PRODUCT=OSC, KEY=19315-00000-2$#0$-4A#49
SYGWLKEY TYPE=ENTRY, PRODUCT=DB2, KEY=26875-00239-2$$$A-#AR#H
SYGWLKEY TYPE=FINAL
```

Building a global customization module (SYGWXCPH)

The installation process in Chapter 3, “Installation and Configuration,” creates 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 these macros:

- SYGWMCST
- SYGWMCXL
- SYGWDRIV
- SYGWHOST
- SYGWLKEY
- TDSGLOB, a relocatable object module

Translation Tables

Topic	Page
Understanding the ASCII-EBCDIC and EBCDIC-ASCII translation tables	45
Default ASCII_8 translation tables	47
Default ISO_1 translation tables	50
Default cp437 (code page 437) translation tables	53
Default cp850 (code page 850) translation tables	55

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 data conversion only if Unicode support is disabled and USEIBMUNICODE=N.

The four pairs of default tables are:

- `ascii_8`
- `iso_1`
- `cp437`
- `cp 850`

Note The `ascii_8` default table also provides the “base” for `roman8` (HP), `ibmascii`, `mac` (Macintosh Roman), and user-definable character sets.

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 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 48.

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

	0	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
0 <u>_</u>	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0C	0E	0F
1 <u>_</u>	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
2 <u>_</u>	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 “ASCII_8, ASCII-to-EBCDIC translation table” on page 48.

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

Setting Up the CICS Sockets Interface

Topic	Page
Understanding the CICS sockets interface	59
Installing and configuring the Client Option of CICS sockets interface	60

Understanding the CICS sockets interface

Mainframe Connect Client Option for CICS contains modules for implementing IBM CICS sockets interface. Before running Mainframe Connect Client Option for CICS, make sure that a CICS region has the IBM CICS sockets interface installed and configured. For details, refer to *z/OS Communications Server: IP CICS Sockets Guide*.

For Mainframe Connect Client Option for CICS, using the IBM CICS sockets interface requires:

- Installing IBM and Sybase RDO definitions for the required programs and transactions
- Concatenating the sockets load library
- Enabling CICS sockets interface automatically or by using the EZAO command

Note To run the Client Option for CICS, you do not need the SYBTPSEC configuration module.

Installing and configuring the Client Option of CICS sockets interface

Follow these procedures to install and configure the CICS sockets interface for Mainframe Connect Client Option for CICS.

❖ **To install and configure the CICS sockets interface**

- 1 Add the IBM CICS sockets RDO entries to your CICS region. You can find input for DFHCSDUP in *tcphlq.SEZAINST(EZACICCT)*.

Note *tcphlq* refers to the high-level qualifier of your system's IBM TCP/IP configuration.

- 2 Define the EZACONFG data set to contain the configuration data for CICS sockets and load the system configuration parameters.

You can find IBM-supplied JCL for this procedural step in *tcphlq.SEZAINST(EZACICFG)*.

- 3 Add *tcphlq.SEZATCP* into the DFHRPL concatenation for the CICS region.

To control the CICS sockets interface, you can use the EZAC and the EZAO commands. To start or stop the entire sockets interface, use EZAO.

Note If you want the CICS sockets interface to start automatically when CICS is initialized, you must add the EZACIC20 program to the second stage of the start-up PLT and to the first stage of the shutdown PLT. Any sockets defined with IMMEDIATE=YES are automatically started.

Setting Up Secure Sockets Layer Protocol

Topic	Page
Understanding Secure Sockets Layer	61
Setting up SSL in Sybase products	63
Setting up SSL in IBM z/OS	65
AT-TLS support in Client and Server Options for CICS	70
For more information	78

Understanding Secure Sockets Layer

Mainframe Connect Client Option and Server Option for CICS supports Secure Sockets Layer (SSL) session-based security. SSL is the standard for securing the transmission of sensitive information (such as credit card numbers, stock trades, and banking transactions) over the Internet. These sections describe the Secure Sockets Layer (SSL) protocol and how it works.

Description of SSL features

SSL provides these features:

- Authentication for clients and servers, with practical emphasis on the server
- Data confidentiality (encryption)
- Verification that a transaction was sent by the client and that the identical transaction was received by the server

To provide efficient authentication and encryption, SSL combines private-key and public-key technologies.

Public-key
cryptography
(asymmetric)

Public-key (asymmetric) cryptography is based on the Public Key Infrastructure (PKI) method of encryption, in which two different keys are used for encrypting and decrypting operations: one is public, the other is private. This means that an operation encrypted by one key can only be decrypted by the other key. The result is that public keys can be seen by all, yet privacy is still possible.

The highly-used RSA algorithm works for both encryption and decryption operations. This method solves critical key exchange issue, but the algorithms it uses require large key sizes and often result in slow CPU-bound operations.

Private-key
cryptography
(symmetric)

In private-key (symmetric) cryptography, the sender and receiver use the same key for both encryption and decryption operations on the same data. Key size is very important: the longer the key, the stronger it is. Currently, 1024, 2048 is the recommended length. The private-key (symmetric) method is 1000 times faster than the public-key (asymmetric) method.

How SSL provides security

This section describes how SSL provides and signs certificates to provide security.

Authentication and
encryption

SSL starts with a “handshake,” during which the client authenticates the server, and the server optionally authenticates the client. Handshake negotiations are based on the public-key cryptography: The client and server agree on how to encrypt and decrypt data, such as using cipher suites and session keys. Also, the format to transmit encrypted data is defined in the handshake.

Certificate Authorities
(CA)

When securing communications, both client and server use X.509 certificates. The client must verify the server's certificate based on the certificate of the Certificate Authority (CA) that signed the certificate or based on a self-signed certificate from the server. (The client verification is optional.) Then, the client and the server use the negotiated session keys and begin encrypted communication, using private-key cryptography.

These are the main fields in a X.509 V3 certificate:

- Issuer
- Subject fields (an X.500 Distinguished Name, commonName field)
- Algorithm identifier
- Subject's public key
- Period of validity

Generating a certificate

- Extensions
- Digital signature

The process of generating a certificate involves these basic steps:

- 1 Generate the public-private key pair.
- 2 Store the private key securely (as password-encrypted).
- 3 Generate a certificate signing request (CSR) in PKCS#10 format for the server certificate.
- 4 Present the CSR to the CA.
- 5 Receive the signed certificate from the CA.
- 6 Store the certificate.

You can use different tools to generate test (self-signed) certificates.

Setting up SSL in Sybase products

As of version 12.5, Sybase added SSL support to some Sybase products, such as ASE, Open Client, Open Server, Replication Server, and jConnect. Currently, some Sybase products use Certicom SSLPlus and Security Builder libraries.

Note Be aware that ASE 12.5 uses SSLPlus, versions 3.1.4/5 and 5.1.4, but ASE 15.0 uses SSLPlus, version 5.1.4.

ASE can use SSL on CIS and Site-handler. Also, ASE and Replication Server require licensing to use SSL.

Some Sybase products have SSL configuration options and can be easily configured to use SSL. Others provide special APIs or require some programming. The following section provides instructions for specific setups.

Setting up SSL in ASE and Open Client

Follow these procedures to configure and enable ASE and Open Client with SSL.

❖ **To set up SSL in ASE**

- 1 Obtain an ASE license for the SSL feature, either ASE_ASM or ASE_SECDIRS (SySAM).
- 2 Obtain a certificate for the server (plus any CA).
Use the Sybase-provided utilities `certreq` and `certauth` to respectively generate and sign server certificates or CA (self-signed) certificates.
- 3 Install the server certificate with the private key appended. The default location and naming scheme for the server certificate is `$SYBASE/$SYBASE_ASE/certificates/<server>.cert`.

Note The `<server>` name must match the server name from the interfaces file, as well as the `CommonName` from the server certificate.

- 4 Install CA certificates for the server at this location:
`$SYBASE/$SYBASE_ASE/certificates/<server>.txt`.
- 5 Use the `sp_ssladmin addcert` command to let the server know about the certificate location.
- 6 To enable SSL in the server, issue this statement:

```
sp_configure "enable ssl", 1
```
- 7 Modify directory services so that listening ports use SSL by adding the keyword `ssl` to the appropriate interfaces file entries.

❖ **To set up SSL in Open Client**

- 1 Install client copies of CA certificates. You must concatenate your `<server>.txt` file to the `trusted.txt` file, or simply create a new `trusted.txt` file with the `<server>.txt` contents.

Client default locations are:

- For UNIX: `$SYBASE/config/trusted.txt`
 - For Windows: `%sybase%\ini\trusted.txt`
- 2 Enable SSL on the port in Directory Servers, for example, the interfaces file.

Setting up SSL in IBM z/OS

Note The term SSL is used to describe both SSL and TLS protocols.

SSL protocol runs above the TCP/IP protocol and below higher-level protocols such as HTTP.

IBM SSL support runs as part of the TCP/IP stack under UNIX System Services (USS).

These levels of authentication and encryption are available with TLS/SSL security:

- Server authentication only
- Client authentication level 1
- Client authentication level 2
- Client authentication level 3

Note For consistency with other Sybase products, Sybase implements server authentication only.

For server authentication to work, the server must have a private key and associated server certificate in the server key database file. To manage the keys and certificates needed for SSL support, you can use the `gskkyman` utility (provided by the System SSL) or RACF Common Keyring support. The server certificate and the CA certificates are stored in a key ring, also called a key database.

Some considerations when using RACF:

- References to RACF apply to any other System Authorization Facility (SAF)-compliant security products that contain the required support.
- For RACF support, all key rings and certificates are stored in the RACF database. There are no separate key database or stash files.

IBM provides the mainframe applications with two options for implementing SSL support:

- System SSL, which runs on top of the TCP/IP stack and provides interfaces to write both client and server applications.

- Application Transparent - Transport Layer Security (AT-TLS), which provides application-transparent secured connections for both client and server. Internally, it uses System SSL interfaces.

The following subsections describe each of these options.

Using System SSL on z/OS

System SSL provides APIs associated with either SSL environment layer or a secure socket connection layer.

- The SSL environment layer defines the general attributes of the environment (such as database file name, timeout, and so on).
- Secure socket connection layer defines the attributes associated with each secure connection. In addition, secure socket connection layers has read and write function calls.

First, the SSL application must create the SSL environment layer. Then, one or more secure socket connection layers can be associated with the SSL environment.

Each layer has four general function calls:

- open
- attribute_set
- initialize
- close

The open function calls return a handle, either an environment handle or a secure socket connection handle, that must be passed as a parameter on subsequent function calls. Read and write functions are full-duplex; however, only one read and one write call can be in progress at one time for any secure socket connection handle.

In addition to using the SSL programming interfaces in the application, a key database must be created for the SSL application. This key database, which contains certificate information, can be an HFS file built and managed by the gskkyman utility or a RACF key ring.

System SSL uses the Integrated Cryptographic Service Facility (ICSF), if it is available. ICSF provides hardware cryptographic support that will be used instead of the System SSL software algorithms.

For System SSL to use the hardware support, the ICSF-started task must be running, and the application user ID must be authorized to the key and certificate handling resources in the RACF CSFSERV class. RACF can also be used to control access to ICSF services.

Note SSL applications must call SSL APIs from a C program because they are C APIs.

Configuring AT-TLS

AT-TLS consolidates TLS implementation in one location, reducing or eliminating application development overhead, maintenance, and parameter specification. AT-TLS is based on z/OS System SSL, and transparently implements it in the TCP layer of the stack.

Applications that are taking advantage of the AT-TLS can be separated into three different types: basic, aware, and controlling. The type is based on whether the application is aware of the service, and if so, the amount of control that the application is given over the security functions. The `SIOCTLSCTL.ioctl` function call provides the interface for the application to query or control AT-TLS.

Basic application	A basic application is unaware that AT-TLS is encrypting or decrypting data.
Aware application	An aware application is aware of AT-TLS and can query information such as AT-TLS status, partner certificate, and derived RACF user ID without any advanced setting in AT-TLS policy.
Controlling application	A controlling application is aware of AT-TLS and needs to control the secure session. It must have the <code>ApplicationControlled</code> parameter in the AT-TLS policy set to ON.

All of these types of applications send and receive unencrypted text data while encrypted data flows over the network.

Follow this procedure to configure AT-TLS policies.

❖ To configure AT-TLS

- 1 Provide the TCP/IP stack with the AT-TLS policies required to negotiate secure connections.

AT-TLS policies are configured in the Policy Agent (described in the next section) using a set of configuration statements and parameters coded into a flat file. You can create the flat file using one of two methods:

- Using manual configuration, coding all the required statements in an HFS file or MVS data set, or
 - Using z/OS Network Security Configuration Assistant, which is a standalone Windows application that requires no network connectivity or setup. You can download the GUI from the Web site for the Communication Server Family downloadable tool.
- 2 Enable AT-TLS through the TTLS parameter on the TCPCONFIG statement in PROFILE.TCPIP.

When AT-TLS is enabled and a newly established connection is first used, the TCP layer of the stack searches for a matching AT-TLS policy installed from the Policy Agent. If no policy is found, the connection is made without AT-TLS involvement.

Configuring AT-TLS policies in the Policy Agent

The Policy Agent component is responsible for implementing policy decisions that control network security and traffic prioritization for the z/OS environment. When initiated, the Policy Agent reads the configuration files, parses the policies, and stores the policy definitions in the TCP/IP stack, which then operates based on the policies. When the policy rule is true, one set of actions is initiated; when it is false, a different set of actions is initiated.

The Policy Agent main configuration file points to other policy files that contain specific policies for TCP/IP images. It can contain a Tcplmage statement that identifies the z/OS UNIX file or MVS data set that contains the policy to be received by a stack. On its end, the TCP/IP image policy file can contain a TTLSConfig statement that identifies the z/OS UNIX file or MVS data set that contains the AT-TLS policy.

Types of configuration files

There are several types of configuration files:

- Main configuration file, determined by using a standard search order
- Common IPsec configuration files
- Common AT-TLS configuration files
- Image configuration files
- Image AT-TLS configuration files

Rule conditions for a connection

Within the AT-TLS policy file, a TTLSRule statement defines a set of conditions that are compared against the connection being checked. When a match is found, policy look-up stops, and the connection is assigned the actions associated with the rule.

The rule conditions apply to connect parameters as follows:

- LocalAddr
- RemoteAddr
- LocalPortrange
- RemotePortrange
- Jobname
- Userid

Direction and at least one other condition must be specified. The TTLSRule statement can reference up to three action statements. In a simple implementation for AT-TLS, these configuration statements should be defined:

- TTLSGroupAction, which must specify TTLSEnabled=ON. The AT-TLS group action represents a single Language Environment process and enclave, and initializes one instance of the System SSL DLL.
- TTLSEnvironmentAction, which must specify a key ring and the handshake role. The AT-TLS environment action initializes a System SSL environment within the Language Environment process that was created to represent an AT-TLS group action.
- TTLSConnectionAction, which specifies attributes for a subset of connections. It is not required for a simple implementation.

❖ **To start the Policy Agent**

- You can start the Policy Agent, which runs as a UNIX process, using one of two methods:
 - From the z/OS shell, where its executable resides in */usr/lpp/tcpip/sbin*, or
 - As a started task using the PAGENT command on an MVS console. You can find a sample started task procedure for PAGENT in TCPIP.SEZAINST(EZAPAGSP).

Note To start Policy Agent from z/OS, you need security product authorization definition (for RACF or any other product).

Policy Agent search order for configuration file information

The Policy Agent search order for accessing the main configuration file (PAGENT.CONF information) is:

- 1 File or data set specified with the `-c` startup option
- 2 File or data set specified with the `PAGENT_CONFIG_FILE` environment variable
- 3 The `etc/pagent.conf` file

Policy Agent environment variables

These environment variables are used to tailor the Policy Agent to a particular installation:

- `PAGENT_CONFIG_FILE`, which points to the main configuration file or data set
- `PAGENT_LOG_FILE`, which points to the log file
- `PAGENT_LOG_FILE_CONTROL`, which controls the number and size of log files

You might also need to define these:

- `TZ`, which defines the local time zone, even if it is defined in `/etc/profile`
- `LIBPATH`, which points to the dynamic link libraries (DLLs) needed to act as an LDAP client

AT-TLS support in Client and Server Options for CICS

In the CICS sockets implementation, transaction security environments are not visible to AT-TLS support. The CICS job and all its transactions appear to the stack as a single server application. As a result, all AT-TLS policy look-up, System SSL key ring authorization checks, and ICSF private key authorization checks are processed using the identity of the CICS job.

The connection that is established, whether active or passive, can perform SSL handshake processing as either the client or the server. All of the connections established by a single CICS job can share the session ID cache in the SSL environment. The CICS job should use a private key ring with a Server certificate, and the key ring used must contain the chain of the root certificates it needs to validate the Server certificate it presents to the client.

Mainframe Connect Client and Server Options for CICS take advantage of the AT-TLS security support, provided that the following conditions are true:

- The TCP/IP stack supports AT-TLS.
- An AT-TLS Policy configuration matches identifiers of the CICS application that will use it, for example, the status of the application as a listener or a client, the IP addresses, and the ports that will be used for communication.
- Digital certificates and key rings are created for these applications.

Configuring a z/OS client or server system

On each z/OS system where a server or client application is to implement AT-TLS security, you need to perform these basic tasks:

- 1 Create a key ring
- 2 Create Policy Agent files
- 3 Add AT-TLS configuration
- 4 Add statements to the *TTLSSConfig* policy file
- 5 Set up *InitStack* access control
- 6 Enable AT-TLS.

The following subsections present an example of configuration tasks performed to ensure SSL secure communication for the following network participants, all of which use self-signed digital certificates:

- A z/OS CICS server named “CICSDEV1”
- A z/OS CICS client
- A Windows ASE server named “ase1”
- A Windows Open Client client

1. Create a key ring

The task of creating a key ring involves all steps for generating and managing digital certificates. To do so, you use RACF commands. Be sure that you have RACF authority.

Note You need the SPECIAL attribute to issue the RACDCERT command. (GROUP-SPECIAL is not sufficient.)

❖ **To create a key ring**

- 1 To activate certificate and key ring classes, use these commands:
 - SETROPTS CLASSACT(DIGTRING)
 - SETROPTS CLASSACT(DIGTCERT)
- 2 To refresh after you make changes, use these commands:
 - SETROPTS RACLIS(DIGTRING) REFRESH
 - SETROPTS RACLIST(DIGTCERT) REFRESH
- 3 To give access to the appropriate resources, use these commands:
 - RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
 - RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
 - PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY)ID(KGUEOR) ACCESS(READ)
 - PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)ID(KGUEOR) ACCESS(READ)
 - PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY)ID(DFHCICS) ACCESS(CONTROL)
- 4 To define a key ring, generate a self-signed certificate, and connect it to the key ring, use these commands:
 - RADCERT ID(DFHCICS) ADDRING(CICSDEV1)
 - RADCERT ID(DFHCICS) GENCERT SUBJECTSDN(CN('CICSDEV1') OU('MFI') O('MFI') C('US')) WITHLABEL ('CICSDEV1CERT') TRUST SIZE(1024)
 - RADCERT ID(DFHCICS) CONNECT (ID(DFHCICS) RING(CICSDEV1) LABEL('CICSDEV1CE RT') DEFAULT)
 - SETROPTS RACLIST (DIGTRING DIGTCERT) REFRESH
- 5 To ensure the Windows SSL client connection to the “CICSDEV1” server, export the newly-created, self-signed certificate to a data set using this command:

```
RADCERT EXPORT(LABEL('CICSDEV1CERT')) ID(DFHCICS) DSN(CICSDEV1.CERT)
```

This puts the certificate contents into the PS data set called KGUEOR.CICSDEV1.CERT using the default FORMAT(CERTB64) and code page 1047.

- 6 FTP the contents of that file to the Windows client's host with ASCII conversion.
- 7 Paste the certificate to the client's CA list in the `%sybase%\ini\trusted.txt` file.
- 8 Add “,ssl” to the CICSDEV1 server entry in the client's `sql.ini` file.
- 9 To ensure that the CICS client (Client Option for CICS) connects to the SSL server named “ase1,” FTP the “ase1” certificate file from the `$$SYBASE/$$SYBASE_ASE/certificates/ase1.txt` file to the KGUEOR.ASE1CERT data set, with ASCII conversion.
- 10 Add the sent FTP “ase1” CA certificate to the RACF database with a TRUST status using this command:


```
RACDCERT ID(DFHCICS) ADD('KGUEOR.ASE1CERT')
WITHLABEL('ASE1CERT') TRUST
```
- 11 Connect the newly added certificate to the key ring and refresh:


```
RACDCERT ID(DFHCICS) CONNECT (ID(DFHCICS) RING(CICSDEV1)
LABEL('ASE1CERT'))

SETROPTS RACLIST (DIGTRING DIGTCERT) REFRESH
```

Note Be aware of certificates encoding, which is generally in binary or text formats. Binary formats must be transported in their exact binary format, without any conversion. In contrast, text formats (such as Base64) must be transported as text. When transporting for an ASCII system, be sure that the ASCII-to-EBCDIC translation is performed.

2. Create Policy Agent files

Note For all Policy Agent configuration file examples, only the content relevant to the AT-TLS configuring is presented here.

The main TCP/IP configuration file is in `/etc/pagent.conf3`. It contains two Tcplmage statements for defining policies on stacks `TCPIP` and `TCPIPT`:

```
# Path: /etc/pagent.conf3
# This is a sample PAGENT.CONF Policy Agent main configuration file
# applied to stacks TCPIP and TCPIPT
LogLevel 15
```

```
# AT-TLS config needed for TCPIPT stack
TcpImage TCPIPT /etc/TCPIPT.policy # FLUSH PURGE 600 params can be used
TcpImage TCPIP /etc/TCPIP.policy # No AT-TLS policy applied to this stack
```

3. Add the AT-TLS configuration

The policy file for the *TCPIPT* stack, */etc/TCPIPT.policy*, contains a *TTLSSConfig* statement pointing to the AT-TLS configuration file for that stack:

```
# This is a sample TCP/IP image configuration file
# Path: /etc/TCPIPT.policy
# This is a sample TCP/IP image configuration file
# applied to stack TCPIPT
LogLevel 15
TTLSSConfig /etc/pagent_TTLS.conf3
```

4. Add statements to the *TTLSSConfig* policy file

The *TCPIPT* stack configuration file */etc/pagent_TTLS.conf3* has all the AT-TLS configuration statements for that stack:

```
# Path: /etc/pagent_TTLS.conf3
# This is a sample AT-TLS configuration file for
# stack TCPIPT, server CICSDEV1.
# Server port #3042 is defined as SSL and port #3042 - as non-SSL
# Client is to connect to a remote SSL port #6005.
#### Inbound definitions ####
#
#
TTLSSRule                DEV1SY02    # Listener on port #3042
{
LocalPortRange           3042
Userid                   DFHCICS
Direction                Inbound
TTLSSGroupActionRef     TTLSGRP1
TTLSEnvironmentActionRef TTLSENV1
}

TTLSSGroupAction TTLSGRP1
{
TLSEnabled On
Trace 1                  # Log Errors to TCP/IP job log
}
TTLSEnvironmentAction TTLSENV1
{
HandshakeRole           Server
```

```

EnvironmentUserInstance 1
TTLSSKeyRingParms
{
  Keyring          CICSDEV1    # Refers to RACF database
}
}

# Port #3043 is not AT-TLS configured
TTLSSRule          DEV1SY03 # Listener on port #3043
{
  LocalPortRange   3043
  Userid           DFHCICS
  Direction        Inbound
  TTLSSGroupActionRef  TTLSSGRP2
}

TTLSSGroupAction  TTLSSGRP2
{
  TTLSEnabled Off
  Trace 1          # Log Errors to TCP/IP job log
}

TTLSEnvironmentAction  TTLSENV2
{
  HandshakeRole     Server
  EnvironmentUserInstance 3
  TTLSSKeyRingParms
  {
    Keyring          CICSDEV1    # Refers to RACF database
  }
}
#
#### Outbound definitions  ####
#
TTLSSRule          DEV1Client    # Client
{
  RemotePortRange   6005          # Remote port to connect to
  Userid           DFHCICS
  Direction        Outbound
  TTLSSGroupActionRef  TTLSSGRP3
  TTLSEnvironmentActionRef  TTLSENV3
}
TTLSSGroupAction  TTLSSGRP3
{
  TTLSEnabled On
  Trace 1          # Log Errors to TCP/IP job log
}

```

```
}  
  
TTLSEnvironmentAction TTLSENV3  
{  
  HandshakeRole      Client  
  EnvironmentUserInstance 1  
  TTLSKeyRingParms  
  {  
    Keyring          CICSDEV1    # Refers to RACF database  
  }  
}
```

5. Set up InitStack access control

You need to define the security product authorization for PAGENT.

When using AT-TLS, z/OS will not allow any socket-based applications to start before PAGENT is up and running. This restriction is needed to verify that all the security policies are enforced. However, some essential applications need to start before PAGENT. For these applications, you need to define a resource INITSTACK profile in the SERVAUTH class. The resource name consists of the following parts:

- EZB – the constant.
- INITSTACK – the constant for this resource type.
- *sysname* – the system name.
- *tcpprocname* – the TCP/IP proc name.

When TCPCONFIG TTLS is defined in the initial *TCPIP.PROFILE*, the INITSTACK profile must be defined. Policy Agent—and any socket-based programs it requires—must be given permission to this resource.

Note Be sure that the program name is the name used to invoke the program—*not* the module name.

Most TCP/IP applications are invoked by ALIAS name. This example lists both names:


```

SETROPTS CLASSACT(SERVAUTH)
SETROPTS RACLIST (SERVAUTH)
SETROPTS GENERIC (SERVAUTH)
RDEFINE SERVAUTH EZB.INITSTACK.*.TCP*    UACC(NONE) PERMIT
EZB.INITSTACK.*.TCP* -
        CLASS(SERVAUTH) ID(*) ACCESS(READ) -
        WHEN(PROGRAM(PAGENT,EZAPAGEN))
SETROPTS GENERIC(SERVAUTH) REFRESH
SETROPTS RACLIST(SERVAUTH) REFRESH
SETROPTS WHEN(PROGRAM) REFRESH

```

6. Enable AT-TLS

To enable AT-TLS functionality for a stack, define the TCPCONFIG TTLS parameter in the TCPIP.PROFILE.

7. Run SSL secure connections

After you complete all previous configuration tasks, you are ready to start the Policy Agent. Here is the example of the PAGENT procedure used:

```

//PAGENT  PROC PARS=' -c /etc/pagent.conf3 -l /tmp/pagent.log4 '
//PAGENT  EXEC PGM=PAGENT, REGION=0K, TIME=NOLIMIT,
//  PARM= (' POSIX(ON) ALL31(ON) ',
//        ' ENVAR ("LIBPATH=/usr/lib" ',
//        ' "TZ=MST7MDT6" ) /&PARMS ' )
//STDENV  DD PATH='/etc/pagent.env', PATHOPTS=(ORDONLY)
//SYSPRINT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//CEEDUMP DD SYSOUT=*, DCB=(RECFM=FB, LRECL=132, BLKSIZE=132)

```

where */etc/pagent.env* contains these definitions:

- LIBPATH=/usr/lib
- TZ=MST7MDT6

After PAGENT starts successfully, start the stack TCPIPT and the CICS region called “CICSDEV1” on that stack. If the server called “ase1” is running, you are ready to test your connections.

For more information

Refer to the *IBM z/OS Communication Server: IP Configuration Guide* for more information about these topics:

- For an overview of the policy-based networking and components, see the chapter called “Policy-based Networking.”
- For information on the AT-TLS security application and configuration, see the chapter called “AT-TLS Data Protection.”
- For an overview of using digital certificates, generating them, and handling them, see the appendix called “TLS/SSL Security.”

For detailed information on Policy Agent statements and policy applications, see the chapter called “Policy Agent Statements and Policy Applications” in the *IBM z/OS Communication Server: IP Configuration Reference*.

For more information about using digital certificates, generating them, and handling them, refer to the chapter called “RACF and Digital Certificates” in the *z/OS Security Server RACF Security Administrator Guide*.

For detailed information about using the RACDCERT (RACF Digital Certificate) command, refer to the *z/OS Security Server RACF Command Language Reference*.

For diagnosing AT-TLS or Policy Agent problems, refer to the appropriate chapters in the *z/OS Communication Server: IP Diagnosis Guide*.

For information on using digital certificates with other security products, refer to related documentation.

For more information on how to obtain digital certificates, configure and use SSL connections with the Sybase products, see the appropriate Sybase product documentation.

Glossary

accept	Establishment of a SNA or TCP/IP connection between Mainframe Connect Server Option and Mainframe Connect DirectConnect for z/OS Option.
access service	The named set of properties, used with an access service library, to which clients connect. Each DirectConnect server can have multiple services.
access code	A number or binary code assigned to programs, documents, or folders that allows authorized users to access them.
access service library	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 .
ACSLIB	See access service library .
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.
Adaptive Server Enterprise/Component Integration Services	Includes a variation of ASE that provides a Transact-SQL interface to various sources of external data. Component Integration Services allows ASE to present a uniform view of enterprise data to client applications.
administrative service library	A service library that provides remote management capabilities and server-side support. It supports a number of remote procedures, invoked as RPC requests, that enable remote DirectConnect server management. See also remote procedure call , service library .
ADMLIB	See administrative service library .
Advanced Interactive Executive	The IBM implementation of the UNIX operating system. The RISC System/6000, among other workstations, runs the AIX operating system.
advanced program-to-program communication	Hardware and software that characterize the LU 6.2 architecture and its implementations in products. See also logical unit 6.2 .

AIX	See Advanced Interactive Executive .
AMD2	The component of the Mainframe Connect DB2 UDB Option that allows clients to submit SQL statements to DB2 UDB. It is a CICS transaction that receives SQL statements sent from Mainframe Connect DirectConnect for z/OS Option and submits them to DB2 UDB, using the DB2 UDB dynamic SQL facility. It also receives the results and messages from DB2 UDB and returns them to Mainframe Connect DirectConnect for z/OS Option.
American Standard Code for Information Interchange	The standard code used for information interchange among data processing systems, data communication systems, and associated equipment. The code uses a coded character set consisting of 7-bit coded characters (including a parity check, 8 bits).
API	See application program interface .
APPC	See advanced program-to-program communication .
application program interface	The programming language interface between the user and Mainframe Connect Client Option or Mainframe Connect Server Option. The API for Mainframe Connect Client Option is Client-Library. The API for Mainframe Connect Server Option is Gateway-Library.
ASCII	See American Standard Code for Information Interchange .
ASE	See Adaptive Server Enterprise .
ASE/CIS	See Adaptive Server Enterprise/Component Integration Services .
batch	A group of records or data processing jobs brought together for processing or transmission.
bind	In the Sybase environment, this term has different meanings depending on the context: <ul style="list-style-type: none">• In CICS, it is an SNA command used to establish a connection between LUs, or a TCP/IP call that connects an application to a port on its system.• In DB2 UDB, it compiles the Database Request Module, the precompiler product that contains SQL statements in the incoming request, and produces an access plan, a machine code version of the SQL statements that specifies the optimal access strategy for each statement.• In the mainframe access product set, it establishes a connection between a TRS port and a CICS or IMS region.

bulk copy transfer	A transfer method in which multiple rows of data are inserted into a table in the target database. Compare with destination-template transfer and express transfer .
call level interface	A programming style that calls database functions directly from the top level of the code. Contrast with embedded SQL .
catalog	A system table that contains information about objects in a database, such as tables, views, columns, and authorizations.
catalog RPC	A component of the Mainframe Connect DB2 UDB Option that allows clients to access DB2 UDB system catalogs. It uses an interface compatible with the catalog interface for the ODBC API.
catalog stored procedure	A procedure used in SQL generation and application development that provides information about tables, columns, and authorizations.
character set	A set of specific (usually standardized) characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.
CICS	See Customer Information Control System .
CICS region	The instance of CICS.
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 . Compare with server .
client application	Software responsible for the user interface that sends requests to applications acting as servers. See also client/server .
Client-Library	A library of routines that is part of Mainframe Connect Client Option.
client request	An RPC or language request sent by a client to a server.
client/server	An architecture in which the client is an application that handles the user interface and local data manipulation functions, and the server is an application providing data processing access and management. See also client application .
Client Services Application	A customer-written CICS program initiated on the host that uses the API to invoke the Mainframe Connect Client Option as a client to the Client Option server or to ASE. See also application program interface, Client Services for CICS .

Client Services for CICS	A Sybase host API that invokes the Mainframe Connect Server Option as a client to an access service for DB2 UDB or ASE. See also application program interface, Customer Information Control System, Client Services Application, Mainframe Connect Server Option.
clustered index	An index in which the physical order and the logical (indexed) order is the same. Compare with nonclustered index.
code page	An assignment of graphic characters and control function meanings to all code points.
commit	A process that makes permanent all changes made to one or more database files since the initiation of the application program, the start of an interactive session, or the last commit or rollback operation. Compare with rollback.
Common Programming Interface	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.
configuration file	A file that specifies the characteristics of a system or subsystem.
configuration set	A section into which service library configuration files are divided.
conversion	The transformation between values that represent the same data item but which belong to different datatypes. Information can be lost due to conversion, because accuracy of data representation varies among different datatypes.
connection	A network path between two systems. For SNA, the path connects a logical unit (LU) on one machine to an LU on a separate machine. For TCP/IP, the path connects TCP modules on separate machines.
connection router	A program provided with Mainframe Connect Client Option that directs requests to particular remote servers. Mainframe system programmers use the connection router to define remote servers and server connections to Mainframe Connect Client Option.
Connection Router Table	A memory-resident table maintained by a Mainframe Connect Client Option system programmer that lists servers and the connections that a Client-Library transaction can use to access them.
control section	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 can be a conditional statement or an imperative statement.

conversation-level security	The passing of client login information to the mainframe by TRS when it allocates a conversation.
CSA	See Client Services Application .
CSP	See catalog stored procedure .
cursor	In SQL, a named control structure used by an application program to point to a row of data.
Customer Information Control System	An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.
DASD	See direct access storage device .
data definition statement	An IBM mainframe statement used to relate a name with a file.
data definition language	A language for describing data and data relationships in a database.
data set name	The term or phrase used to identify a data set.
database management system	The term or phrase to identify a data set. A computer-based system for defining, creating, manipulating, controlling, managing, and using databases.
database operation	A single action against the database. For Mainframe Connect DirectConnect for z/OS Option, a database operation is usually a single SQL statement. One or more database actions can be grouped together to form a request. See also request .
Database 2	An IBM relational database management system.
datatype	A keyword that identifies the characteristics of stored information on a computer.
DB-Library	A Sybase and Microsoft API that allows client applications to interact with ODS applications. See also application program interface .
DBMS	See database management system .
DB2 UDB	See Database 2 .
DDL	See data definition language .
DD statement	See data definition statement .
default language	The language that displays a user's prompts and messages.

destination-template transfer	A transfer method in which source data is briefly put into a template where the user can specify that some action be performed on it before execution against a target database. See also transfer . Compare with bulk copy transfer and express transfer .
direct access storage device	A device in which access time is effectively independent of the location of the data.
direct request	A request sent directly from a client workstation through Transaction Router Service to the DirectConnect server without going through ASE. Contrast with indirect request .
direct resolution	A type of service name resolution that relies upon a client application specifying the exact name of the service to be used. See also service name resolution . Compare with service name redirection .
DirectConnect Manager	A Java application from Sybase that can be used in Windows and UNIX environments. It provides remote management capabilities for DirectConnect products, including starting, stopping, creating, and copying services.
Client Option server	The component of Mainframe Connect DirectConnect for z/OS Option that provides general management and support functions to service libraries.
dll	See dynamic link library .
DSN	See data set name .
dynamic link library	A file containing executable code and data bound to a program at load time or runtime, rather than during linking.
dynamic SQL	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. Contrast with static SQL .
ECDA	See Enterprise Connect Data Access .
ECDA Option for ODBC	A Sybase solution that allows client applications to access ODBC data. It combines the functionality of the ECDA Option for ODBC architecture with ODBC to provide dynamic SQL access to target data, as well as the ability to support stored procedures and text and image pointers.
ECDA Option for Oracle	A Sybase solution that provides Open Client access to Oracle databases. When used in combination with ASE, it provides many of the features of a distributed database system, such as location transparency, copy transparency, and distributed joins.

embedded SQL	SQL statements that are embedded within a program and are prepared in the 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.
end user	A person who connects to a DirectConnect server using an application to access databases and perform transfers. See also transfer .
Enterprise Connect Data Access	An integrated set of software applications and connectivity tools that allow access to data within a heterogeneous database environment, such as a variety of LAN-based, non-Sybase data sources, as well as mainframe data sources.
environment variable	A variable that describes how an operating system runs and the devices it recognizes.
exit routine	A user-written routine that receives control at predefined user exit points.
express transfer	A form of bulk copy transfer that uses ODBC bulk APIs to improve performance when transferring bulk data between data sources. Because it uses the same syntax as bulk copy transfer, no modification of applications is required.
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.
External Security Manager	An add-on security package for the z/OS mainframe, licensed by Computer Associates.
FCT	See forms control table .
forms control table	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.
Gateway-Library	A library of communication, conversion, tracing, and accounting functions supplied with Mainframe Connect Server Option.
globalization	The combination of internationalization and localization. See 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 .

handler	A routine that controls a program's reaction to specific external events, for example, an interrupt handler.
host	The mainframe or other machine on which a database, an application, or a program resides. In TCP/IP, this is any system that is associated with at least one Internet address. See also Transmission Control Protocol/Internet Protocol .
host ID	In Mainframe Connect Server Option, the ID that the TRS passes to the mainframe with a client request. The host ID is part of the client login definition at the TRS.
host password	In Mainframe Connect Server Option, the password that the client passes to the mainframe with a client request.
host request library	A DB2 UDB table that contains host-resident SQL statements that can be executed dynamically. See also host-resident request .
host-resident request	A SQL request that resides in a DB2 UDB table called the host request library. See also host request library .
IMS	See Information Management System .
indirect request	A client request that is routed through a stored procedure on a SQL Server, which forwards the request to TRS as an RPC. Compare with direct request .
Information Management System	A database/data communication system that can manage complex databases and networks.
interfaces file	An operating system file that determines how the host client software connects to a Sybase product. An <i>interfaces</i> file entry contains the name of any Client Option server and a list of services provided by that server.
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 can be localized for a specific culture. See also globalization . Compare with localization .
keyword	A word or phrase reserved for exclusive use by Transact-SQL.
language RPC	The name TRS uses to represent a client's language request. TRS treats a language request as a remote procedure call (RPC) and maps it to a language transaction at the remote server.

language transaction	The server transaction that processes client language requests. The Mainframe Connect DB2 UDB Option language transaction for CICS is AMD2, which uses the DB2 UDB dynamic SQL facilities to process incoming SQL strings. The Mainframe Connect DB2 UDB Option for IMS uses SYRT by default.
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 that creates load modules from one or more object modules or creates load modules by resolving cross references among the modules, and if necessary, adjusts those addresses.
link-edit	To create a loadable computer program by using a linkage editor. See also linkage editor .
localization	The process of preparing an extracted module for a target environment, in which 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 .
local variable	A variable that is defined and used only in one specified portion of a computer program. Contrast with global variable .
logical unit	A type of network addressable unit that enables a network user to gain access to network facilities and communicate remotely. A connection between a TRS and a CICS region is a connection between logical units.
logical unit 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 .
login ID	In Mainframe Connect Server Option, the ID that a client user uses to log in to the system.
login packet	Client information made available to Mainframe Connect Server Option. The client program sets this information in a login packet and sends it to TRS, which forwards it to the mainframe.
long-running transaction	A transaction that accepts more than one client request. Whereas short transactions end the communication after returning results to a client, a long-running transaction can await and process another request. Compare with short transaction .
LU 6.2	See logical unit 6.2 .

mainframe access products	Sybase products that enable client applications to communicate with mainframes in a client/server environment. See client/server .
Mainframe Connect	The Sybase product set that provides access to mainframe data.
Mainframe Connect Client Option	A Sybase product that, using Client-Library, allows mainframe clients to send requests to SQL Server, Open Server, the Mainframe Connect DB2 UDB Option and Mainframe Connect Server Option. Mainframe Connect Client Option provides capability for the mainframe to act as a client to LAN-based resources in the CICS or the IMS and MVS environment.
Mainframe Connect DB2 UDB Option	A Sybase mainframe solution that provides dynamic access to DB2 UDB data. It is available in the CICS or IMS environment. See also Customer Information Control System, Database 2, Multiple Virtual Storage .
Mainframe Connect Client Option for z/OS Option	A Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management.
Mainframe Connect Server Option	A Sybase product that provides capability for programmatic access to mainframe data. It allows workstation-based clients to execute customer-written mainframe transactions remotely. It is available for the CICS and the IMS and MVS environments
Multiple Virtual Storage	An IBM operating system that runs on most System/370 and System/390 mainframes. It supports 24-bit addressing up to 16 megabytes.
network protocol	A set of rules governing the way computers communicate on a network.
nonclustered index	An index that stores key values and pointers to data. Compare with clustered index .
null	Having no explicitly assigned value. NULL is not equivalent to 0 or to blank.
ODBC	See Open Database Connectivity .
ODS	See Open Data Services .
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 Data Services	A product that provides a framework for creating server applications that respond to DB-Library clients.
Open Database Connectivity	A Microsoft API that allows access to both relational and non-relational databases. See also application program interface .

Open Server	A Sybase product that provides the tools and interfaces required to create a custom server. Clients can route requests to the Client Option server through an Open Server configured to meet specific needs, such as the preprocessing of SQL statements.
parameter	A variable that is given a constant value for a specified application and can denote the application. Compare with property .
Partner Certification Reports	Sybase publications that certify third-party or Sybase products to work with other Sybase products.
Password Expiration Management	An IBM password management program with CICS Version 3.3 through an optional program temporary fix, and as an integral part of CICS with version 4.1 and higher.
PEM	See Password Expiration Management .
PL/1	See Programming Language /1 .
primary database	The database management system that the DirectConnect server is always connected to. It is implied in the transfer statement.
Programming Language/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. Compare with parameter .
protocol	The rules for requests and responses used to manage a network, transfer data, and synchronize the states of network components.
query	A request for data from a database, based upon specified conditions.
Registry	The part of the Windows operating system that holds configuration information for a particular machine.
relational database	A database in which data is viewed as being stored in tables consisting of columns (data items) and rows (units of information).
relational operators	Operators supported in search conditions.
relops	See relational operators .
remote procedure call	A call to execute a stored procedure on a remote server. For Mainframe Connect Server Option, an RPC is a direct request from a client to TRS. For Mainframe Connect Client Option, a Client-Library transaction that calls a procedure on a remote server acts like an RPC.

remote stored procedure	A customer-written CICS program using an API that resides on the mainframe and communicates with Mainframe Connect DB2 UDB Option. See also Customer Information Control System, stored procedure . Compare with Client Services Application .
remote systems management	A feature that allows a system administrator to manage multiple DirectConnect servers and multiple services from a client.
Replication Server	A Sybase SQL Server application that maintains replicated data and processes data transactions received from a data source.
request	One or more database operations an application sends as a unit to the database. Depending upon the response, the application commits or rolls back the request. See also commit, rollback, unit of work .
resource table	A main storage table that associates each resource identifier with an external logical unit (LU) or application program.
rollback	An instruction to a database to back out of changes requested in a unit of work. Compare with commit .
router	An attaching device that connects two LAN segments, which use similar or different architectures, at the Open System Interconnection (OSI) reference model network layer. Contrast with gateway .
RPC	See remote procedure call .
RSP	See remote stored procedure .
SAA	See System Application Architecture .
secondary connection	The connection specified in the transfer statement. It represents anything that can be accessed using Mainframe Connect Client Option, such as ASE or another access service.
secondary database	In transfer processing, the supported database that is specified in the transfer statement. Compare with primary database .
server	A functional unit that provides shared services to workstations over a network. See also client/server . Compare with client .
server process ID	A positive integer that uniquely identifies a client connection to the server.
service	A functionality available in Mainframe Connect DirectConnect for z/OS Option. It is the pairing of a service library and a set of specific configuration properties.

service library	In Mainframe Connect DirectConnect for z/OS Option, a set of configuration properties that determine service functionality. See also access service library , administrative service library , Transaction Router Service library , transfer service library .
service name redirection	A type of service name resolution that allows a system administrator to create an alternative mechanism to map connections with services. See also service name resolution . Compare with direct resolution .
service name redirection file	The default name of the file used for the service name redirection feature. See service name redirection .
service name resolution	The DirectConnect server mapping of an incoming service name to an actual service. See also direct resolution , service name redirection .
session	A connection between two programs or processes. In APPC communications, sessions allow transaction programs to have conversations between the partner LUs. See also advanced program-to-program communication .
short transaction	A mainframe transaction that ends the communication when it finishes returning results to the client. Compare with long-running transaction .
SNA	See Systems Network Architecture .
SNRF	See service name redirection file .
SPID	See server process ID .
SQL	See structured query language .
SQLDA	See SQL descriptor area .
sqledit	A utility for creating and editing <i>sql.ini</i> files and file entries.
sql.ini	The interfaces file containing definitions for each Client Option server to which a workstation can connect. The file must reside on every client machine that connects to ASE.
SQL descriptor area	A set of variables used in the processing of SQL statements.
SQL stored procedure	A single SQL statement that is statically bound to the database. See also stored procedure .
static SQL	SQL statements that are embedded within a program and prepared during the program preparation process before the program runs. Compare with dynamic SQL .

stored procedure	A collection of SQL statements and optional control-of-flow statements stored under a particular name. Adaptive Server stored procedures are called “system procedures.” See also remote stored procedure, system procedures.
structured query language	An IBM industry-standard language for processing data in a relational database.
stub	A program module that transfers remote procedure calls (RPCs) and responses between a client and a server.
SYRT	The component of Mainframe Connect DB2 UDB for IMS that allows clients to submit SQL language requests to DB2 through IMS.
System Administrator	The person in charge of server system administration, including installing and maintaining DirectConnect servers and service libraries.
System Application Architecture	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.
system procedures	A stored procedure that ASE supplies for use in system administration. System procedures serve as shortcuts for retrieving information from system tables, or a mechanism for accomplishing database administration. See also stored procedure.
Systems Network Architecture	An IBM proprietary plan for the structure, formats, protocols, and operational sequences for transmitting information units through networks. See also advanced program-to-program communication.
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	A Sybase application-level protocol that defines the form and content of relational database requests and replies.
target	A system, program, or device that interprets, rejects, satisfies, or replies to requests received from a source.
target database	The database to which the DirectConnect server transfers data or performs operations on specific data.
TCP/IP	See Transmission Control Protocol/Internet Protocol.
TDS	See Tabular Data Stream.

transaction	A unit of processing initiated by a single request. A transaction consists of one or more application programs that, when executed, accomplish a particular action. In Mainframe Connect Server Option, a client request (RPC or language request) invokes a mainframe transaction. In Mainframe Connect Client Option, a mainframe transaction executes a stored procedure on a remote server.
transaction processing	A sequence of operations on a database that is viewed by the user as a single, individual operation.
Transaction Router Service	A Mainframe Connect DirectConnect for z/OS Option program used when the mainframe acts as a transaction server to route requests from remote clients to the Mainframe Connect Server Option and return results to the clients.
Transaction Router Service library	A service library that facilitates access to remote transactions, allowing customers to execute transactions from virtually any mainframe data source. See also service library .
Transact-SQL	A Sybase-enhanced version of the SQL database language used to communicate with ASE.
transfer	A Mainframe Connect DirectConnect for z/OS Option feature that allows users to move data or copies of data from one database to another.
transfer service library	A service library that provides copy management functionality. See also service library .
Transmission Control Protocol/Internet Protocol	A set of communication protocols that supports peer-to-peer connectivity functions for both local and wide area networks.
trigger	A form of stored procedure that automatically executes when a user issues a change statement to a specified table.
TRS	See Transaction Router Service .
TRS library	See Transaction Router Service library .
T-SQL	See Transact-SQL .
unit of work	One or more database operations grouped under a commit or rollback. A unit of work ends when the application commits or rolls back a series of requests, or when the application terminates. See also commit , rollback , transaction .
user ID	User identification. The ID number by which a user is known in a specific database or system.

variable	An entity that is assigned a value. Mainframe Connect Client Option for z/OS Option has two kinds of variables: <i>local</i> and <i>global</i> .
view	An alternate representation of data from one or more tables. A view can include all or some of the columns contained the table or tables on which it is defined.
Virtual Storage Access Method	An IBM-licensed program that controls communication and the flow of data in an SNA network.
Virtual Telecommunications Access Method	IBM mainframe software that allows communication on an SNA network between mainframes and allows the mainframe to have multiple sessions per connection.
VSAM	See Virtual Storage Access Method .
VTAM	See Virtual Telecommunications Access Method .
wildcard	A special character that represents a range of characters in a search pattern.

Index

A

- ACCESSCODE SYGWCST parameter 30
- ACCESSCODESW SYGWCST parameter 30
- ASCII_8 translation tables
 - ASCII_8 ACSII-to-EBCDIC 48
 - ASCII_8 EBCDIC-to-ASCII 48
- ASE
 - setting up SSL 64
- asymmetric cryptography 62
- AT-TLS
 - adding statements to the TLSConfig policy file 74
 - adding the configuration file 74
 - configuring 67
 - configuring with z/OS client or server 71
 - creating a key ring 71
 - enabling 77
 - rule conditions for a connection 69
 - secure connections 66
 - setting up InitStack access control 76
 - support in Client and Server Options for CICS 70
 - types of configuration files 68
- authentication levels in SSL 65
- authorization key
 - ordering 15

B

- binary formats
 - in certificate encoding 73
- browsing
 - setting with isql 24

C

- CEDA window 40
- certificate encoding 73

- certificates
 - activating for z/OS 72
 - generating 63
 - installing client copies 64
 - key database 66
- CHARSETSRV SYGWCST parameter 30
- choosing
 - dynamic network drivers 3, 7
 - network drivers 3, 7
- CICS
 - network driver 39
- CICS region
 - installing CICS sockets interface 60
- CICS sockets interface 59
 - description 59, 61
 - EZACIC20 program 60
 - EZACONFG data set 60
 - installing and configuring 59, 60, 61
 - RDO entries 60
 - setting up 59, 61
- Client Option
 - communication at the mainframe 6
 - communication at the server 6
 - network configuration 2
 - security 7
- commands
 - exporting new certificate to a data set 72
 - EZAO 60
 - open 66
 - TCPCONFIG 68
 - to connect a certificate to a key ring 72
 - to define a key ring 72
 - to generate a self-signed certificate 72
 - to refresh after changes to key ring 72
- commands to activate certificate and key ring classes 72
- concatenating
 - sockets load library 60
- configuration
 - creating Policy Agent files 73

Index

- configuration files
 - AT-TLS 68
- configuring
 - AT-TLS 67
 - CICS sockets interface 59, 60, 61
 - tasks for AT-TLS with z/OS 71
 - z/OS client or server for AT-TLS 71
- connections
 - AT-TLS security 66
 - AT-TLS security rules 69
- connectivity
 - verifying gateway-less 13
 - verifying two-tier 13
- cp437 translation tables
 - cp437 ASCII-to-EBCDIC 54
 - cp437 EBCDIC-to-ASCII 55
- cp850 translation tables
 - cp850 ASCII-to-EBCDIC 57
 - cp850 EBCDIC-to-ASCII 58
- CPI-C CICS network driver 39
- creating
 - key ring for AT-TLS 71
 - SSL environment layer 66
- customization options
 - SYGWDRIV 39
 - SYGWHOST, TCP/IP configuration macro 41
- customization table SWGWXCPH 29
 - SYGWMCST global macro 29
 - SYGWMCXL, character set macro 29
- customizing
 - a network driver 39
 - global 30
 - LAN-side character sets 41
 - options 29

D

- DEBUGSW SYGWMCST parameter 30
- DECPOINT SYGWMCST parameter 31
- defining
 - key ring for AT-TLS with z/OS 72
- defining
 - dynamic network drivers 2
 - network drivers 2
 - TCP/IP Server-Host Mapping Table 7

- DQUOTETRAN SYGWMCST parameter 31
- dynamic network driver
 - choosing 3, 7
 - CICS 39
 - CPI-C CICS 39
 - customizing 39
 - defining 2
 - invoking 2
 - loading 2
 - macro 39
 - network type and environment 3

E

- enabling tracing 24
- encryption
 - levels in SSL 65
 - SSL 62
- entering SQL commands 26
- environment
 - gateway-less 2
 - two-tier 2
- environment variables
 - Policy Agent 70
- Establish Open Client Server Connection window 25
- EZACIC20 program
 - CICS sockets interface 60
- EZACONFG data set 60
- EZAO command 60

F

- function key listing
 - isql 25

G

- gateway-less
 - description 2
 - verifying connectivity 13
- generating
 - certificates 63
 - self-signed certificate 72

global customization (SYGWMCST) 29
 list of parameters 30

I

IMSLOGTYPE SYGWMCST parameter 31
 Initstack
 setting up for access control in AT-TLS 76
 installing
 CICS sockets interface 59, 60, 61
 Integrated Cryptographic Service Facility (ICSF) 66
 invoking
 dynamic network drivers 2
 network drivers 2
 ISO_1 translation tables
 ISO_1 ASCII-to-EBCDIC 51
 ISO_1 EBCDIC-to-ASCII 51
 isql
 about 6
 browsing 24
 tracing 24
 viewing results 27

K

key database
 certificate information 66
 key ring
 defining AT-TLS for z/OS 72
 keyboard definitions
 isql functions 25

L

LAN-side character sets
 customizing 41
 LIBPATH 70
 library names
 using new 13
 licensing key
 temporary 15
 loading
 dynamic network drivers 2

network drivers 2
 LONGVARTRUNC SYGWMCST parameter 31

M

macros
 SWGWHOST 41
 SYGWDRIV 3, 39
 mainframe character set customization options
 (SYGWMCXL) 33
 overriding SBCS translation tables 34
 SBCS, customizing translation 35
 SBCS, predefined character sets 34
 SBCS, user-defined character sets 34
 MVSDDDNAME SYGWMCST parameter 32

N

NATLANGUAGESRV SYGWMCST parameter 32
 network communication definitions
 choosing a network driver 3
 overview 3
 network driver
 choosing 3, 7
 CICS 39
 CPI-C CICS 39
 customizing 39
 defining 2
 invoking 2
 loading 2
 macro 39
 network type and environment 3
 network type and environment
 dynamic network driver 3
 network driver 3

O

Open Client
 installing client copies of CA certificates 64
 setting up SSL 64
 open command 66
 ordering a permanent authorization key 15

P

PAGENT_CONFIG_FILE 70
 PAGENT_LOG_FILE 70
 PAGENT_LOG_FILE_CONTROL 70
 parameters
 SYGWMCST macro 30
 Policy Agent
 creating configuration files 73
 environment variables 70
 environment variables 70
 search order for configuration files 70
 starting 69, 77
 private-key cryptography 62
 public-key cryptography 62

R

RDO entries for CICS sockets interface 60
 reference information
 for SSL 78
 result tracing 27
 ROWLIMIT SYGWMCST parameter 32

S

sample network diagram 5
 SBCS translation tables 45
 ASCII_8 47
 cp437 51
 cp850 55
 ISO_1 50
 SBCS translation, customizing
 overriding defaults with SYGWMCXL 36
 Secure Sockets Layer. *See* SSL
 security 7
 for Client Option 7
 Secure Sockets Layer 7
 using SSL 61
 Server-Host Mapping table
 about 6
 setting up
 CICS sockets interface 59, 61
 SSL in z/OS 65
 sockets load library

 concatenating 60
 SQL command entry 26
 SQL commands
 entering 26
 SSL
 authentication and encryption 62
 authentication and encryption levels 65
 creating an environment layer 66
 description 61
 features 61
 private-key cryptography 62
 public-key cryptography 62
 reference information 78
 running secure connections with Policy Agent 77
 setting up in ASE 64
 setting up in IBM z/OS 65
 setting up in Open Client 64
 System SSL 65
 X.509 certificates 62
 starting
 Policy Agent 69
 SYGMCST 29
 SYGWDRIV macro 3, 39
 SYGWHOST
 formats 41
 macro 41
 SYGWMCST parameters 30
 ACCESSCODE 30
 ACCESSCODESW 30
 CHARSETSRV 30
 DEBUGSW 30
 DECPOINT 31
 DQUOTETRAN 31
 IMSLOGTYPE 31
 LONGVARTRUNC 31
 MVSDDNAME 32
 NATLANGUAGESRV 32
 ROWLIMIT 32
 USEIBMUNICODE 32
 SYGWMCXL 33
 SYGWXCPH 3
 definitions 7
 symmetric cryptography 62
 System SSL
 using on z/OS 66

T

- tasks
 - configuration for AT-TLS with z/OS 71
- TCP/IP
 - using the SYGWHOST macro 7
- TCPCONFIG command 68
- temporary licensing key 15
- Trace/Error Display results 27
- tracing
 - enabling 24
 - setting with isql 24
- translation tables, SBSCS 45
- trusted.text file 64
- two-tier
 - description 2
 - verifying connectivity 13
- TZ 70

U

- UNIX System Services (USS) 65
- USEIBMUNICODE SYGWMCSST parameter 32

V

- viewing trace results 27

W

- windows
 - CEDA 40
 - Establish Open Client Server Connection 25

X

- X.509 certificates
 - in SSL 62

Z

- z/OS
 - setting up SSL 65
 - using System SSL 66

