

SYBASE®

Utility Guide

**Adaptive Server® Enterprise**

15.0.2

[ Windows and UNIX ]

DOCUMENT ID: DC30191-01-1502-01

LAST REVISED: November 2008

Copyright © 2008 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.

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.

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

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

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

<b>About This Book .....</b>	<b>ix</b>	
<b>CHAPTER 1</b>	<b>Building Servers Using dataserver .....</b>	<b>1</b>
	Introduction .....	1
	Building a new master device .....	2
	Environments when using dataserver .....	3
	build mode .....	3
	start mode .....	6
	Upgrading to a server with larger page sizes .....	7
	Viewing the current server limits .....	7
<b>CHAPTER 2</b>	<b>Using Interactive isql from the Command Line.....</b>	<b>9</b>
	Before you begin .....	9
	Starting and stopping isql .....	9
	How to use Transact-SQL in isql.....	10
	Formatting isql output.....	11
	Correcting input .....	12
	set options that affect output .....	12
	Changing the command terminator.....	13
	Performance statistics interaction with command terminator values .....	14
	Setting the network packet size .....	15
	Input and output files.....	15
	UNIX command-line redirection .....	15
<b>CHAPTER 3</b>	<b>Using Interactive SQL in Graphics Mode.....</b>	<b>17</b>
	Starting Interactive SQL .....	18
	Main window description .....	19
	Plan dialog tab.....	19
	Using the Interactive SQL toolbar .....	21
	Opening multiple windows .....	21
	Keyboard shortcuts .....	21
	Using Interactive SQL to display data .....	22
	Editing table values in Interactive SQL .....	23

- Copying rows from the Interactive SQL result set ..... 24
- Editing table values from the Interactive SQL result set..... 24
- Inserting rows into the database from the Interactive SQL result set  
25
- Deleting rows from the database using Interactive SQL ..... 25
- Working with SQL statements in Interactive SQL ..... 26
  - Canceling an Interactive SQL command..... 26
  - Combining multiple statements ..... 27
  - Looking up tables, columns, and procedures..... 28
  - Recalling commands ..... 29
  - Logging commands ..... 29
- Configuring Interactive SQL ..... 30
  - General dialog box ..... 30
  - Result dialog box..... 31
  - Import/Export dialog box ..... 32
  - Messages dialog tab ..... 33
  - Editor ..... 33
  - Query Editor ..... 34
- Running command files..... 35
  - Writing output to a file..... 35
  - Executing command files ..... 35
  - Saving, loading, and running command files..... 36
- Using SQL escape syntax in Interactive SQL ..... 37
- Interactive SQL commands ..... 38

**CHAPTER 4**

**Using bcp to Transfer Data to and from Adaptive Server..... 41**

- Methods for moving data..... 42
  - Importing and exporting data with bcp ..... 42
- bcp requirements ..... 43
- bcp modes..... 45
- Copying data to a file ..... 45
  - Examples..... 45
- bcp performance ..... 46
  - Using fast or slow bcp ..... 47
  - Copying in data with fast bcp ..... 50
  - Bulk copying data into partitioned tables..... 50
  - Using parallel bulk copy to copy data into a specific partition . 53
- Using the bcp options..... 60
  - Using the default formats ..... 60
  - Changing terminators from the command line ..... 61
- Changing the defaults: interactive bcp ..... 62
  - Responding to bcp prompts ..... 63
  - File storage type..... 64
  - Prefix length ..... 66

Field length.....	67
Field and row terminators.....	69
Using format files.....	72
Elements of the bcp format file.....	73
Examples: copying out data interactively.....	76
Copying out data with field lengths.....	77
Copying out data with delimiters.....	78
Examples: copying in data interactively.....	79
Copying in data with field lengths.....	80
Copying in data with delimiters.....	82
Copying in data with a format file.....	82
Using bcp with alternate languages.....	83
bcp and row-level access rules.....	84
Copy in and batch files.....	84
Improving recoverability.....	85
Batches and partitioned tables.....	86
Copy out and text and image data.....	86
Specifying a network packet size.....	87
Copy in and error files.....	87
Copy out and error files.....	88
Data integrity: defaults, rules, and triggers.....	89
Defaults and datatypes.....	89
Rules and triggers.....	89
How bcp differs from other utilities.....	89

**CHAPTER 5**

<b>Using dsedit.....</b>	<b>91</b>
Getting started with dsedit.....	91
Starting dsedit.....	91
Opening an editing session.....	93
Adding, viewing, and editing server entries.....	95
Modifying server entries in Windows.....	95
Modifying server entries in UNIX platforms.....	98
Copying server entries.....	100
Troubleshooting dsedit.....	102
The dsedit utility does not start.....	102
Error message: "Unable to open X display".....	102
Cannot add, modify, or delete server entries.....	103

**CHAPTER 6**

<b>Using dscp.....</b>	<b>105</b>
Getting started with dscp.....	105
Using a dscp session.....	106
Working with server entries.....	107
Adding and modifying server entries.....	107

Copying server entries .....	109
Listing and viewing contents of server entries.....	110
Deleting server entries .....	111
Exiting dscp .....	112
Quick reference for dscp utility commands .....	112

<b>CHAPTER 7</b>	<b>Utility Commands Reference .....</b>	<b>115</b>
	Getting started.....	116
	*_r utilities.....	118
	Utilities quick reference .....	118
	Installation or configuration utilities .....	119
	Utilities for languages, character sets, and sort orders .....	119
	Utilities to start servers .....	120
	Database creation and manipulation utilities .....	120
	Utilities to gather information.....	121
	backupserver.....	122
	bcp .....	128
	buildmaster.....	142
	certauth .....	143
	certpk12 .....	147
	certreq .....	150
	charset .....	153
	cobpre .....	154
	cpre .....	155
	dataserver .....	156
	ddlgen .....	163
	defncopy.....	186
	dscp.....	192
	dsedit.....	193
	extractjava .....	194
	installjava .....	197
	isql.....	201
	langinstall .....	213
	optdiag .....	216
	preupgrade .....	223
	pwdcrypt.....	228
	showserver .....	229
	sqlbgr .....	230
	sqlloc .....	235
	sqllocres .....	236
	sqlsrvr.....	237
	sqlupgrade .....	244
	sqlupgraderes .....	245
	srvbuild.....	246

srvbuildres .....	248
startserver.....	249
sybmigrate .....	252
xpserver.....	257

## CHAPTER 8

<b>Migration Utility .....</b>	<b>259</b>
Overview.....	259
Benefits of sybmigrate .....	259
What sybmigrate does .....	260
What sybmigrate does not do .....	262
Before you begin .....	263
Required components for the sybmigrate.....	263
Dependencies.....	263
Installation.....	264
Permissions .....	264
Platforms.....	265
Environment settings .....	265
Migrating proxy tables.....	265
Migration process .....	266
Overview of the migration process .....	266
Pre-migration considerations.....	267
Configuration and tuning for higher performance .....	269
Possible errors to avoid .....	272
Auto-select dependent objects for migration .....	272
Migrating an archive database.....	272
Starting sybmigrate.....	273
GUI mode .....	277
Resource file mode.....	284
Sybmigrate with Encrypted Columns.....	291
Post-migration activities.....	292
Migrating databases in the Replication Server domain .....	293
Premigration procedures .....	293
Postmigration procedures.....	294
Migrating databases that support wide data .....	299
Limitations .....	300
Troubleshooting and error messages.....	302
Objects fail to migrate .....	302
Beginning database migration .....	302
“Connection refused” and “Unable to obtain connection to the server” .....	302
Target server cannot be reached from source server.....	303
If sybmigrate hangs during migration.....	303
Merging two databases.....	303
Post-migration failure cleanup .....	304

---

Remigrating one database.....	304
Re-creating an individual object.....	304
Connection fail .....	305
“Insufficient memory in JVM shared class” .....	305
“There is not enough memory in the procedure cache” .....	305
java.lang related error .....	305
<b>Index.....</b>	<b>307</b>



# About This Book

*Adaptive Server Enterprise Utility Guide* is a guide to the Sybase® Adaptive Server® Enterprise utility programs available for UNIX platforms and Windows. Utility programs are commands that you invoke directly from the operating system.

## **Audience**

This manual is for anyone using Transact-SQL® and Adaptive Server Enterprise version 15.0. It assumes that you have the basic knowledge to use Adaptive Server and your operating system.

## **How to use this book**

This manual includes the following:

- Chapter 1, “Building Servers Using dataserver” – discusses how to use the `dataserver` utility to build new servers.
- Chapter 2, “Using Interactive `isql` from the Command Line” – discusses how to use the interactive SQL (`isql`) utility that allows access to SQL from your operating system.
- Chapter 3, “Using Interactive SQL in Graphics Mode” – describes a how to use the GUI-based Interactive SQL utility interactive SQL to execute SQL statements, build scripts, and display database data to the server.
- Chapter 4, “Using `bcp` to Transfer Data to and from Adaptive Server” – discusses, in detail, the bulk copy (`bcp`) utility which you use to move data between Adaptive Server and an operating system file.
- Chapter 5, “Using `dsedit`” – explains how to use the directory services editor (`dsedit`) utility to modify the interfaces (`sql.ini`) file in Windows, and in X-Windows to view and edit server entries in the interfaces file in UNIX platforms.
- Chapter 6, “Using `dscp`” – explains how to use the `dscp` utility to view and edit server entries in the interfaces file in UNIX platforms.
- Chapter 7, “Utility Commands Reference” – lists and describes the utility commands that you use to manage and maintain your databases and Adaptive Server Enterprise.

- 
- Chapter 8, “Migration Utility” – explains how to use the sybmigrate utility to move data and database schema from pre-12.5 databases into 15.0 databases.

The examples in this manual are based on the pubs2 sample database. Ask your System Administrator how to access a clean copy of pubs2.

#### Related documents

The Sybase® Adaptive Server® Enterprise documentation set consists of the following:

- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.

A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What’s New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 15.0, the system changes added to support those features, and changes that may affect your existing applications.
- *ASE Replicator User’s Guide* – describes how to use the Adaptive Server Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User’s Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- The *Configuration Guide* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *Full-Text Search Specialty Data Store User’s Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User’s Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.

- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as data types, functions, and stored procedures in the Adaptive Server database.
- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Messaging Service User's Guide* – describes how to use Real Time Messaging Services to integrate TIBCO Java Message Service and IBM WebSphere MQ messaging services with all Adaptive Server database applications.
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books for Adaptive Server that explain how to tune Adaptive Server for maximum performance:
  - *Basics* – the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.
  - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
  - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL<sup>®</sup> information:
  - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  - *Commands* – Transact-SQL commands.

- 
- *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.
  - *Tables* – Transact-SQL system tables and dbcc tables.
  - *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
  - *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
  - *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
  - *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
  - *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
  - *Unified Agent and Agent Management Console* – Describes the Unified Agent, which provides runtime services to manage, monitor and control distributed Sybase resources.
  - *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
  - *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.
  - *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
  - *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

**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.

v **Finding the latest information on product certifications**

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click Certification Report.
- 3 In the Certification Report filter select a product, platform, and time frame and then click Go.
- 4 Click a Certification Report title to display the report.

v **Finding the latest information on component certifications**

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.

- 
- 2 Either select the product family and product under Search by Product; or select the platform and product under Search by Platform.
  - 3 Select Search to display the availability and certification report for the selection.

v **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**

v **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 EBFs/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBFs/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 EBFs/Maintenance report, or click the product description to download the software.

**Conventions**

The following sections describe conventions used in this manual.

SQL is a free-form language. There are no rules about the number of words you can put on a line or where you must break a line. However, for readability, all examples and most syntax statements in this manual are formatted so that each clause of a statement begins on a new line. Clauses that have more than one part extend to additional lines, which are indented. Complex commands are formatted using modified Backus Naur Form (BNF) notation.

Table 1 shows the conventions for syntax statements that appear in this manual:

**Table 1: Font and syntax conventions for this manual**

Element	Example
Command names, procedure names, utility names, and other keywords display in sans serif font.	<code>select</code> <code>sp_configure</code>
Database names and datatypes are in sans serif font.	<code>master database</code>
Book names, file names, variables, and path names are in italics.	<i>System Administration Guide</i> <i>sql.ini</i> file <i>column_name</i> \$SYBASE/ASE directory
Variables—or words that stand for values that you fill in—when they are part of a query or statement, are in italics in Courier font.	<code>select column_name</code> <code>from table_name</code> <code>where search_conditions</code>
Type parentheses as part of the command.	<code>compute row_aggregate (column_name)</code>
Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates “is defined as”.	<code>::=</code>
Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.	<code>{cash, check, credit}</code>
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	<code>[cash   check   credit]</code>
The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.	<code>cash, check, credit</code>
The pipe or vertical bar ( ) means you may select only one of the options shown.	<code>cash   check   credit</code>
An ellipsis (...) means that you can <b>repeat</b> the last unit as many times as you like.	<code>buy thing = price [cash   check   credit]</code> <code>[, thing = price [cash   check   credit]]...</code> You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.

- 
- Syntax statements (displaying the syntax and all options for a command) appear as follows:

```
sp_dropdevice [device_name]
```

For a command with more options:

```
select column_name  
from table_name  
where search_conditions
```

In syntax statements, keywords (commands) are in normal font and identifiers are in lowercase. Italic font shows user-supplied words.

- Examples showing the use of Transact-SQL commands are printed like this:

```
select * from publishers
```

- Examples of output from the computer appear as follows:

pub_id	pub_name	city	state
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

(3 rows affected)

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

Adaptive Server's sensitivity to the case of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see the *System Administration Guide*.

## Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.



Adaptive Server 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.



# Building Servers Using *dataserver*

Adaptive Server version 15.0 no longer uses the buildmaster binary to build the master device. Instead, Sybase has incorporated the buildmaster functionality in the *dataserver* binary. This chapter discusses how to use *dataserver* to build your server.

Topic	Page
Introduction	1
Building a new master device	2

---

**Note** The *dataserver* binary in Windows is called *sqlsrvr.exe*. If you are using the Windows platform, substitute all reference to *dataserver* in this chapter with *sqlsrvr*.

---

For a detailed description of *dataserver* syntax, see *dataserver* on page 156. For a detailed description of *sqlsrvr* syntax, see *sqlsrvr* on page 237.

## Introduction

The *dataserver* command allows you to create master devices and databases with logical pages of size 2K, 4K, 8K, or 16K. Larger logical pages allow you to create larger rows, which can improve your performance because Adaptive Server accesses more data each time it reads a page. For example, a 16K page can hold eight times the amount of data as a 2K page, an 8K page holds four times as much data as a 2K page, and so on, for all the sizes for logical pages.

The logical page size is a server-wide setting; you cannot have databases with varying size logical pages within the same server. All tables are appropriately sized so that the row size does not exceed the current page size of the server. That is, rows cannot span multiple pages.

## Building a new master device

This section describes the process for creating a new master device using the `dataserver` utility. The master device is built using the **build** mode in `dataserver`. After the master device is built, the server shuts down. You must then manually start the server in the **start** mode. After this you can start, stop, and restart Adaptive Server whenever necessary without having to rebuild the master device

---

**Note** When you are building a master device you should allow an additional 8K for the config block.

---

Adaptive Server uses three types of page sizes:

- Logical page size – these are the pages that the database objects are built with. A databases and any of its related objects must use the same logical page size. Logical page sizes come in sizes of 2K, 4K, 8K, and 16K.
- Virtual page size – this is the physical page allocation at the disk level, and is always done in 2K pages. All disk I/O is done in multiples of virtual page size.
- Memory page size – the memory allocated and managed within Adaptive Server. The memory page size is always in units of 2K pages.

The following syntax creates a new master device with `dataserver`:

```
dataserver -ddevice_name
. . .
-b [master_device_size [k|K|m|M|g|G]
[-z logical_page_size [k|K]
-h
```

Where:

`-d device_name` – is the full path name of the device for the master database. The master database device must be writable by the user who starts Adaptive Server. The default master database device name is `d_master`.

`-b` – indicates that `dataserver` is in build mode and creating a new master device, and indicates the size of the master device. If you do not provide a unit specifier (k, m, g) for the size of the device, `dataserver` assumes a size in virtual pages. The size of a virtual page is always 2K. For example:

- `-b 51204` – specifies a device of 51,204 virtual pages (100.0078125MB).
- `-b 100M` – specifies a device of 100MB

**-z** – specifies the logical page size, which is always 2K, 4K, 8K, or 16K. This parameter is optional during the build phase and is ignored during the start mode. If you do not include the **-z** parameter during the build mode, the master device is built with 2K logical pages.

**-h** – prints the syntax for the *dataserver* command.

See *dataserver* on page 156 for a full list of *dataserver* parameters and their definitions.

## Environments when using *dataserver*

When you start an Adaptive Server with the *dataserver* program, Adaptive Server derives its running environment from:

- The configuration file you specify in **-c *configuration\_file***
- The default configuration file, *servername.cfg*, if you did not specify the **-c** parameter
- Default values if you did not specify either **-c *configuration\_file*** or *servername.cfg*

For more information on these configuration parameters, see Chapter 17, “Setting Configuration Parameters,” in the *System Administration Guide*.

## build mode

To create a new Adaptive Server, issue *dataserver* using the **-b** and **-z** options. For example, to:

- Build a 100MB master device using the default logical page size (2K) and start the server:

```
dataserver -d /var/sybase/masterdb.dat -b100M -sMASTER2K
```

- Build a 100MB master device with a logical page size of size 4K:

```
dataserver -d /var/sybase/masterdb.dat -b100M -z4K -sMASTER4K
```

- Build a master device of 102,400 virtual pages of size 2K, create databases using a logical page size of 8K, and boot the server:

```
dataserver -d /var/sybase/masterdb.dat -b102400 -z8K -sMASTER8K
```

If the total requested space (102,400 x 2K = 200 MB) is insufficient to build all the required system databases using the specified logical page size, then an error message is reported, and the process fails.

**Example**

The following is a sample output of `dataserver` building a 200MB device with a 2K logical page size, called `personnel2k`:

```
dataserver -d /var/sybase/personnel2k.dat -b200M -z2k -sPERSONNEL2K
```

`dataserver` uses a default configuration file if you do not specify one:

```
00:00000:00000:2001/04/16 10:24:31.73 kernel Warning: Using default file
'/var/sybase/PERSONNEL2K.cfg' since a configuration file was not specified.
Specify a configuration file name in the RUNSERVER file to avoid this
message.
```

To specify your own configuration file, use the `dataserver -c` parameter. See Chapter 11, “Setting Configuration Parameters” in the *System Administration Guide* for more information.

Adaptive Server version 15.0 treats all installations as an upgrade, regardless of whether you have an existing version of Adaptive Server or not. For this reason, you see the following output when running `dataserver`:

```
00:00000:00001:2001/04/16 10:24:32.63 server Database 'master' appears to
be at an older revision than the present installation; SQL Server will assess
it, and upgrade it as required.
```

```
00:00000:00001:2001/04/16 10:24:32.66 server Database 'master': beginning
upgrade step [ID 1]: Initialize disk and create empty allocation units
on master device.
```

```
00:00000:00001:2001/04/16 10:24:34.74 server Database 'master': beginning
upgrade step [ID 2]: Bootstrap basic system catalogs in database.
```

`dataserver` continues creating the master database, including all of its tables such as `systypes`, `sysobjects` and `sysusages`:

```
00:00000:00001:2001/04/16 10:24:35.21 server Database 'master': beginning
upgrade step [ID 3]: creating index (table systypes, index ncsystypes)
```

```
00:00000:00001:2001/04/16 10:24:35.36 server Database 'master': beginning
upgrade step [ID 4]: creating index (table sysobjects, index
ncsysobjects)
```

```
00:00000:00001:2001/04/16 10:24:35.44 server Database 'master': beginning
upgrade step [ID 20]: creating table (table sysusages)
```

```
[...]
```

When `dataserver` has created the master database, it creates the model database:

[...]

```
00:00000:00001:2001/04/16 10:24:43.14 server Database 'model' appears to
be at an older revision than the present installation; SQL Server will assess
it, and upgrade it as required.
```

```
00:00000:00001:2001/04/16 10:24:43.14 server Database 'model': beginning
upgrade step [ID 1]: Initialize disk and create empty allocation units
on master device.
```

```
00:00000:00001:2001/04/16 10:24:43.83 server Database 'model': beginning
upgrade step [ID 2]: Bootstrap basic system catalogs in database.
```

```
00:00000:00001:2001/04/16 10:24:43.89 server Database 'model': beginning
upgrade step [ID 3]: creating index (table systypes, index ncsystypes)
```

```
00:00000:00001:2001/04/16 10:24:43.91 server Database 'model': beginning
upgrade step [ID 4]: creating index (table sysobjects, index
ncsysobjects)
```

[...]

When `dataserver` has created the model database, it creates the `tempdb` and `sysystemdb` databases:

[...]

```
00:00000:00001:2001/04/16 10:24:45.23 server CREATE DATABASE: allocating
1024 logical pages (2.0 megabytes) on disk 'master'.
```

```
00:00000:00001:2001/04/16 10:24:46.79 server Database sysystemdb
successfully created.
```

[...]

`dataserver` is successful when the server changes the default sort order and shuts down:

[...]

```
00:00000:00001:2001/04/16 10:24:47.23 server Now loading SQL Server's new
default sort order and character set
```

[...]

```
00:00000:00001:2001/04/16 10:24:47.31 server Default Sort Order
successfully changed.
```

```
00:00000:00001:2001/04/16 10:24:47.37 server  SQL Server shutdown after
verifying System Indexes.
```

```
00:00000:00001:2001/04/16 10:24:47.37 kernel  ueshutdown: exiting
```

### Error messages

If `dataserver` is not successful, you cannot boot the server on that master device, and you see the following error message:

```
00:00000:00001:2001/04/16 19:02:39.53 kernel  Use license file
/var/sybase/SYSAM-1_0/licenses/license.dat.
```

```
00:00000:00001:2001/04/16 19:02:39.54 kernel  The master device's
configuration area appears to be corrupt. The server needs this data to boot,
and so cannot continue. The server will shut down.
```

If you run `dataserver` with a user-specified configuration file that includes options that make it impossible to allocate a shared segment and start up a server, `dataserver` fails with an error message, and you cannot boot the server on that master device:

```
00:00000:00001:2001/04/16 19:04:01.11 kernel  Use license file
/var/sybase/SYSAM-1_0/licenses/license.dat.
```

```
00:00000:00000:2001/02/09 19:04:01.25 kernel  Using config area from primary
master device.
```

```
00:00000:00001:2001/04/16 19:04:01.36 server  The value of the 'max
total_memory' parameter (33792) defined in the configuration file is not high
enough to set the other parameter values specified in the configuration file.
'max total_memory' should be greater than the logical memory '34343'.
```

## start mode

To start an existing Adaptive Server, issue `dataserver` without the `-b` and `-z` options.

```
dataserver -d /sybase/masterdb.dat
```



## Upgrading to a server with larger page sizes

Adaptive Servers earlier than version 12.5 used 2K logical page sizes. You cannot change an installation's page size by upgrading. That is, if your current Adaptive Server uses 2K logical pages, you can upgrade only to an Adaptive Server that uses 2K logical pages.

However, you can migrate databases with 2K logical pages from earlier versions of Adaptive Server. For information on how to use the `sybmigrate` data migration tool, see Chapter 8, "Migration Utility."

## Viewing the current server limits

To display information about Adaptive Server's limits:

- `dbcc serverlimits` includes the size of your server's logical page size in its output. For example, enter:

```
dbcc serverlimits
```

- Search for the string "logical page size" in the error log.
- The global variable `@@maxpagesize` displays the server's logical page size. At the `isql` prompt, issue:

```
select @@maxpage size
-----
      8192
```



# Using Interactive isql from the Command Line

This chapter describes the command line SQL utility, isql.

<b>Topic</b>	<b>Page</b>
Before you begin	9
Starting and stopping isql	9
How to use Transact-SQL in isql	10
Changing the command terminator	13
Performance statistics interaction with command terminator values	14
Setting the network packet size	15
Input and output files	15

For a detailed description of isql syntax, see isql on page 201.

## Before you begin

If you are running Open Client version 11.1 or later and are using an external Sybase configuration file, you must add the following in your configuration file to enable isql:

```
[isql]
```

## Starting and stopping isql

To start isql, enter this command at the operating-system prompt:

```
isql
```

When the prompt appears, enter your password.

The password does not appear on the screen as you type. The isql prompt appears:

```
1>
```

You can now issue Transact-SQL commands.

To exit isql enter either of these commands on a line by itself:

```
quit  
exit
```

## How to use Transact-SQL in isql

isql sends Transact-SQL commands to Adaptive Server, formatting the results and printing them to standard output. There is no maximum size for an isql statement. For more information about using Transact-SQL, see the *Transact-SQL User's Guide*.

---

**Note** To use Transact-SQL directly from the operating system with the isql utility program, you must have an account, or login, on Adaptive Server.

---

To execute a Transact-SQL command, type the default command terminator “go” on a new line.

For example:

```
isql  
Password:  
  
1> use pubs2  
2> go  
1> select *  
2> from authors  
3> where city = "Oakland"  
4> go
```

## Formatting isql output

The width for isql output is adjusted according to the character set expansion or the character width, and displays a output column of the maximum possible bytes. For example, for the UTF8 character set, each character may use at most four bytes, so the output column width is the character number multiplied by four. However, the output column width can not be larger than the column defined value, and the column width is calculated using this formula:

$$\text{Min}(\text{character\_number} \times \text{max\_character\_width}, \text{column\_defined\_width}).$$

For example, if a column `co11` is defined as `varchar(10)`, then `left(co11, 2)` returns a width of eight, or four bytes per character. A value of `left(co11,5)` returns a width of 10, and cannot be larger than the defined length, even though, according to the formula,  $5 \times 4$  equals 20).

Table 2-1 describes the options that change the format of isql output:

**Table 2-1: Format options for isql**

Option	Default	Meaning
<code>-h headers</code>	1	Number of rows to print between column headings
<code>-s colseparator</code>	Single space	Changes the column separator character
<code>-w columnwidth</code>	80 characters	Changes the line width
<code>-e</code>		Includes each command issued to isql in the output
<code>-n</code>		Removes numbering and prompt symbols.

In this example, the query's results are placed in a file called *output*:

```
isql -User_name -Ppassword -Sserver -e -n -o output
use pubs2
go
select *
from authors
where city = "Oakland"
go
quit
```

To view the contents of *output*, enter:

- In Windows:
 

```
type output
```
- In UNIX platforms:
 

```
cat output
```

```
select *
```

```
from authors
where city = "Oakland"
```

```
au_id      au_lname  au_fname  phone      address
  city      state country  postalcode
-----
213-46-8915 Green      Marjorie  415 986-7020 309 63rd St. #411
Oakland    CA    USA    94618
274-80-9391 Straight   Dick      415 834-2919 5420 College Av.
Oakland    CA    USA    94609
724-08-9931 Stringer   Dirk      415 843-2991 5420 Telegraph Av.
Oakland    CA    USA    94609
724-80-9391 MacFeather Stearns   415 354-7128 44 Upland Hts.
Oakland    CA    USA    94612
756-30-7391 Karsen     Livia     415 534-9219 5720 McAuley St.
Oakland    CA    USA    94609
```

---

**Note** The *output* file does not include the command terminator.

---

## Correcting input

If you make an error when typing a Transact-SQL command, you can:

- Press Ctrl-c or type the word “reset” on a line by itself – this clears the query buffer and returns the isql prompt.
- Type the name of your text editor on a line by itself – this opens a text file where you can edit the query. When you write and save the file, you are returned to isql and the corrected query appears. Type “go” to execute it.

## set options that affect output

Table 2-2 lists the set options that affect Transact-SQL output. For more information, see *set* in the *Reference Manual*.

**Table 2-2: set options that affect Transact-SQL output**

set option	Default	Meaning
char_convert	Off	Turns character-set conversion off and on between Adaptive Server and a client; also starts a conversion between the server character set and a different client character set.

set option	Default	Meaning
fipsflagger	Off	Warns when any Transact-SQL extensions to entry-level SQL92 are used. This option does not disable the SQL extensions. Processing completes when you issue the non-ANSI SQL command.
flushmessage	Off	Sends messages as they are generated.
language	us_english	Sets the language for system messages.
nocount	Off	Turns off report of number of rows affected.
noexec	Off	Compiles each query but does not execute it; often used with showplan.
parseonly	Off	Checks the syntax of queries and returns error messages without compiling or executing the queries.
showplan	Off	Generates a description of the processing plan for a query; does not print results when used inside a stored procedure or trigger.
statistics io statistics time	Off	Displays performance statistics after each execution.
statistics subquerycache	Off	Displays the number of cache hits, misses, and rows in the subquery cache for each subquery.
textsize	32K	Controls the number of bytes of text or image data returned.

## Changing the command terminator

If you include the command terminator argument (-c), you can choose your own terminator symbol; go is the default value for this option. Always enter the command terminator without blanks or tabs in front of it.

For example, to use a period as the command terminator, invoke isql as follows:

```
isql -c .
```

A sample isql session with this command terminator looks like this:

```
1> select name from sysusers
2> .
name
-----
sandy
kim
leslie
(3 rows affected)
```

Using the isql command terminator option with scripts requires advance planning:

- Adaptive Server supplies scripts, such as installmaster, use “go”. Do not change the command terminator for any session that uses these scripts.
- Your own scripts may already have “go” in them. Remember to update your scripts to include the terminator you plan to use.

## Performance statistics interaction with command terminator values

isql provides a performance statistics option (-p).

For example, this syntax returns the following statistics:

```
isql -p
1> select * from sysobjects
2> go

Execution Time (ms.): 1000    Clock Time (ms.): 1000
1 xact:
```

This means that a single transaction took 100 milliseconds. The clock time value reflects the entire transaction, which starts when Client-Library™ builds the query and ends when Client-Library returns the information from Adaptive Server.

You can gather performance statistics based on the execution of one or more transactions. To gather statistics on more than one transaction, specify a number after the command terminator.

For example, the following command instructs Adaptive Server to execute three select \* transactions and report the performance statistics:

```
isql -p
1> select * from sysobjects
2> go 3

Execution Time (ms.): 1000    Clock Time (ms.): 1000
Execution Time (ms.): 1000    Clock Time (ms.): 2000
Execution Time (ms.): 1000    Clock Time (ms.): 1000

Execution Time (ms.): 1000    Clock Time (ms.): 4000
3xact:
```



## Setting the network packet size

Setting the correct network packet size can greatly increase the performance of Adaptive Server.

The `-A` size option specifies the network packet size to use for an isql session. The default value is 2048 bytes. To set the packet size to 4096 bytes for the current isql session, enter:

```
isql -A 4096
```

To check your network packet size, type:

```
select * from sysprocesses
```

The value for this isql session appears under the `network_pktsize` heading in the `sysprocesses` table.

See the *System Administration Guide* for more information about setting the network packet size.

## Input and output files

You can specify input and output files on the command line with the `-i` and `-o` options.

isql does not provide formatting options for the output. However, you can use the `-n` option to eliminate the isql prompts and other tools to reformat the output.

If you use the `-e` option, isql echoes the input to output. The resulting output file contains both the queries and their results.

## UNIX command-line redirection

The UNIX redirection symbols, “<” and “>”, provide a similar mechanism to the `-i` and `-o` options, as follows:

```
isql -Usa < input > output
```

You can direct isql to take input from the terminal, as shown in this example:

```
isql -Usa -Ppassword -Sserver_name << EOF > output
use pubs2
go
```

```
select * from table
go
EOF
```

“<<EOF” instructs isql to take input from the terminal up to the string “EOF.” You can replace “EOF” with any character string. Similarly, the following example signals the end of input with Ctrl-d:

```
isql -Usa << > output
```

# Using Interactive SQL in Graphics Mode

This chapter discusses how to run and use the GUI-based Interactive SQL utility.

<b>Topic</b>	<b>Page</b>
Starting Interactive SQL	18
Main window description	19
Using the Interactive SQL toolbar	21
Opening multiple windows	21
Keyboard shortcuts	21
Using Interactive SQL to display data	22
Editing table values in Interactive SQL	23
Working with SQL statements in Interactive SQL	26
Configuring Interactive SQL	30
Running command files	35
Using SQL escape syntax in Interactive SQL	37
Interactive SQL commands	38

Interactive SQL allows you to execute SQL statements, build scripts, and display database data to the server. You can use it to:

- Browse the information in a database.
- Test SQL statements that you plan to include in an application.
- Load data into a database and carrying out administrative tasks.

In addition, Interactive SQL can run command files or script files. For example, you can build repeatable scripts to run against a database and then use Interactive SQL to execute these scripts as batches.

## Starting Interactive SQL

The menu item Open Interactive SQL opens a connection to a server. However, when you select the menu item for a server, Interactive SQL opens a connection to the default database for that server. When you select a specific database from the Open Interactive SQL menu, Interactive SQL opens to the selected database.

### v **Starting Interactive SQL from Sybase Central**

- To start Interactive SQL, either:
  - Select a database in Sybase Central and select File | Open Interactive SQL. Interactive SQL connects to the database. You can also right-click on the database and select Open Interactive SQL.
  - To start Interactive SQL without a connection to a server, select Tools | Adaptive Server Enterprise | Open Interactive SQL. The Connect dialog appears.

### v **Starting Interactive SQL from the command line**

How you start Interactive SQL from the command line depends on your operating system.

- If you start Interactive SQL independently, the Connect dialog appears, which lets you connect to a database just as you would in Sybase Central.
  - For UNIX, Sybase Central and Interactive SQL do not have to be installed under `$SYBASE` (`$SYBASE` does not even need to exist for them to start). Instead, they are installed under `$SYBROOT`, an environment variable set by the installer. Move to `$SYBROOT` and enter:

```
dbisql
```

On Windows, change to the `%SYBROOT` directory and enter:

```
dbisql.bat
```

For the Mac OS, open the Interactive SQL application under the `/Applications/Sybase` folder.

- In the Connection dialog, enter the information to connect to a database in the Connect dialog box and click OK.

To open a new Interactive SQL window:

- 1 Choose Window | New Window. The Connect dialog appears.
- 2 In the Connect dialog, enter connection options, and click OK to connect.

The connection information (including the database name, your user ID, and the database server) appears on the title bar above the SQL Statements pane.

You can also connect to or disconnect from a database with the Connect and Disconnect commands in the SQL menu, or by executing a connect or disconnect statement in the SQL Statements pane.

## Main window description

The Interactive SQL window includes these panes:

- SQL Statement – provides a place for you to type SQL statements.
- Results – displays the results of commands that you execute. For example, if you use SQL statements to search for specific data in the database, the Results tab in this pane displays the columns and rows that match the search criteria. If the information exceeds the size of the pane, scroll bars automatically appear. You can edit the result set on the Results tab.
- Messages – displays messages from the database server.
- Plan – displays the query optimizer's execution plan for a SQL statement. For more information, see Plan dialog tab below.

The window title displays the connection name. For Adaptive Server, the connection name is either the server name (determined by the server's interfaces file entry) or the host name and port number the user enters at the time of connection.

## Plan dialog tab

The Plan tab displays a GUI representation of execution engine's plan for the currently running SQL text, and helps you understand the performance and statistic characteristics of the currently running query.

---

**Note** The Plan tab only appears if you connect to Adaptive Server version 15.0 and later.

---

The top half of the Plan tab shows the logical flow of the operators used in the plan in a tree-based, hierarchical structure, with each operator a separate node of the tree. The cost of each operator is based on the cost model used by the query processor. Each operator node in the tree is costed relative to other nodes, which makes it easier to identify operators based on their costs.

Each node includes tooltip text (text that appears when you move your mouse over the node) that provides details about each operator, so you do not have to select the nodes to compare details between operators.

The Plan tab includes the following tabs:

- Details – shows the details of the operator statistics as two tables:
  - Node Statistics – shown for all the operators, and includes statistics like row count, logical I/O, and physical I/O.
  - Subtree Statistics – the aggregate sum of all the operators below, and are shown for the non-leaf operators, and include statistics on row count, logical I/O, and physical I/O.
- XML – shows the result set as XML output.
- Text – displays the text version of the query plan (the same as the output of showplan).
- Advanced – includes the following:
  - Abstract query plan – shows the abstract query plan used by the query.
  - Resource utilization – describes the resources used by the plan, including number of threads and the auxiliary session descriptors (SDESSs, every table scan requires one session descriptor to track the scan).
  - Cost – lists costs associated with the plan, including logical I/O, Physical I/O, and CPU usage.
  - Optimizer Metrics – lists the query-plan statistics, including the run time for the query, run time for the first plan, the number of plans evaluated, the number of plans that were valid, and the amount of procedure cache used.
  - Optimizer Statistics – lists the last time you ran update statistics on the table, any missing histogram steps, and the density of the steps.

## Using the Interactive SQL toolbar

The Interactive SQL toolbar appears at the top of the Interactive SQL window. Use the buttons on this toolbar to:

- Recall the executed SQL statement immediately before your current position in the history list.
- View a list of up to 50 previously executed SQL statements.
- Recall the executed SQL statement immediately after your current position in the history list.
- Execute the SQL statement currently appearing in the SQL Statements pane.
- Interrupt the execution of the current SQL statement.

## Opening multiple windows

You can open multiple Interactive SQL windows. Each window corresponds to a separate connection. You can connect simultaneously to two (or more) databases on different servers, or you can open concurrent connections to a single database.

## Keyboard shortcuts

Interactive SQL provides the following keyboard shortcuts.

**Table 3-1: Keyboard shortcuts**

Function	Description
ALT+F4	Exits Interactive SQL.
ALT+LEFT CURSOR	Displays the previous SQL statement in the history list.
ALT+RIGHT CURSOR	Displays the next SQL statement in the history list
CTRL+C	Copies the selected row(s) and column headings to the clipboard.
CTRL+END	Moves to the bottom of the current pane.
CTRL+F6	Cycles through the open Interactive SQL windows.
CTRL+H	Displays the history of your executed SQL statements during the current session.
CTRL+HOME	Moves to the top of the current pane.

Function	Description
CTRL+N	Clears the contents of the Interactive SQL window.
CTRL+Q	Displays the Query Editor, which helps you build SQL queries. When you have finished building your query, click OK to export it back into the SQL Statements pane.
CTRL+S	Saves the contents of the SQL Statements pane.
SHIFT+F5	Refreshes the plan without executing the statement in the SQL Statements pane. This allows you to see the plan for a statement without altering table data.
ESC	Clears the SQL Statements pane.
F2	Allows you to edit the selected row in the result set. You can use the Tab key to move from column to column within the row.
F5	Executes all text in the SQL Statements pane. You can also perform this operation by clicking the Execute SQL Statement button on the toolbar.
F7	Displays the Lookup Table Name dialog. In this dialog, you can find and select a table and then press Enter to insert the table name into the SQL Statements pane at the cursor position. Or, with a table selected in the list, press F7 again to display the columns in that table. You can then select a column and press Enter to insert the column name into the SQL Statements pane at the cursor position.
F8	Displays the Lookup Procedure Name dialog. In this dialog, you can find and select a procedure, then press Enter to insert the procedure name into the SQL Statements pane at the cursor position.
F9	Executes the selected text in the SQL Statements pane. If no text is selected, all of the statements are executed.
Page down	Moves down in the current pane.
Page up	Moves up in the current pane.

## Using Interactive SQL to display data

Interactive SQL allows you to browse the information in databases.

You can display database information using the `select` statement in Interactive SQL. Once you enter the statement, click the Execute SQL Statement button on the toolbar.

After you execute the statement, the result set appears in the Results pane. You can use the scroll bars to see areas of the table that are outside your current view of the pane.

v **Listing all the columns and rows of the *authors* table**

- 1 Start Interactive SQL and connect to the pubs2 database.



- 2 Enter the following in the SQL Statements pane:

```
select * from authors
```

- 3 On the toolbar, click the Execute SQL Statement button.  
You can add, delete, and update rows within the result set.

## Editing table values in Interactive SQL

Once you execute a query in Interactive SQL, you can edit the result set to modify the database. You can also select rows from the result set and copy them for use in other applications. Interactive SQL supports editing, inserting, and deleting rows. These actions have the same result as executing update, insert, and delete statements.

Before you can copy, edit, insert, or delete rows, you must execute a query in Interactive SQL that returns a result set on the Results tab in the Results pane. When you edit the result set directly, Interactive SQL creates and executes a SQL statement that makes your change to the database table.

To edit a row or value in the result set, you must have the proper permissions on the table or column you want to modify values from. For example, to delete a row, you must have delete permission for the table the row belongs to.

Editing the result set may fail if you:

- Attempt to edit a row or column you do not have permission on.
- Select columns from a table with a primary key, but do not select all of the primary key columns.
- Attempt to edit the result set of a join (for example, there is data from more than one table in the result set).
- Enter an invalid value (for example, a string in a numeric column or a NULL in a column that does not allow NULLs).

When editing fails, an Interactive SQL error message appears explaining the error, and the database table values remain unchanged.

Once you make changes to table values, you must enter a commit statement to make the changes permanent. To undo your changes, you must execute a rollback statement.

## Copying rows from the Interactive SQL result set

You can copy rows directly from the result set in Interactive SQL and then paste them into other applications. Copying rows also copies the column headings. Copied data is comma-delimited, which allows other applications, such as Microsoft Excel, to format the copied data correctly. By default, copied data is in ASCII format, and all strings are enclosed in single quotes. You can select only consecutive rows in the result set.

To copy rows from the Interactive SQL result set:

- 1 Select the rows you want to copy. To select a row(s):
  - Press and hold the Shift key while clicking the row(s), or
  - Press and hold the Shift key while using the Up or Down arrow.
- 2 Right-click the result set and select Copy from the pop-up menu. You can also copy the selected row(s) by pressing CTRL+C.

The selected row(s), including their column headings, are copied to the clipboard. You can paste them into other applications by selecting Edit | Paste or by pressing CTRL+V.

## Editing table values from the Interactive SQL result set

The Edit command allows you to change individual values within a row. You can change any or all of the values within existing rows in database tables. You must have update permission on the columns being modified. When you edit the result set, you can make changes to the values in only 'one row at a time.

### v **Editing a row in the result set**

- 1 Select the row to edit
- 2 Right-click the result set and choose Edit from the pop-up menu. You can also press F2 to edit the result set.
- 3 A blinking cursor appears in the first value in the row.
- 4 Press Tab to move the cursor from column to column across the row. You can also edit a value by clicking the value in the selected row.
- 5 Enter the new value.
- 6 You cannot enter invalid datatypes into a column. For example, you cannot enter a string datatype into a column that is configured for the int datatype.

- 7 Execute a commit statement to make your changes to the table permanent.

## Inserting rows into the database from the Interactive SQL result set

The Insert command adds a new blank row to the database table. Use the Tab key to move between columns in the result set to add values to the row. When you add values to the table, characters are stored in the same case as they are entered. You must have insert permission on the table to add new rows (for information about insert permissions, see insert in the *Adaptive Server Commands and Reference Manual*).

### v Inserting a new row into the result set

- 1 Right-click the result set and choose Add from the pop-up menu.

A new blank row appears in the result set with a blinking cursor in the first value in the row.

Press Tab to move the cursor from column to column across the row. You can also insert a value by clicking on the appropriate field in the selected row.

- 2 Enter the new value.

You cannot enter invalid datatypes into a column. For example, you cannot enter a string into a column that accepts the int datatype.

- 3 Execute a commit statement to make your changes to the table permanent.

## Deleting rows from the database using Interactive SQL

The Delete command removes the selected rows from a database table. You must have delete permission on the table to delete rows.

### v To delete a row from the result set:

- 1 Select the row(s) you want to delete. To select a rows:
  - Press and hold the Shift key while clicking the rows.
  - Press and hold the Shift key while using the Up or Down arrow.

To delete non-consecutive rows, you must delete each row individually.

- 2 Right-click the result set and choose Delete from the pop-up menu. You can also delete the selected row(s) by pressing the Delete key.  
The selected row(s) are removed from the database table.
- 3 Execute a commit statement to make your changes to the table permanent.

## Working with SQL statements in Interactive SQL

You can enter all SQL statements as commands in the top pane of the Interactive SQL window. When you are finished typing, execute the statement to run it.

To execute a SQL statement, either:

- Press the Execute SQL Statement button, or
- Select F5.

To clear the SQL Statements pane, choose Edit | Clear SQL or press ESCAPE.

---

**Note** If you are running a long-running query, Interactive SQL displays a splash screen that describes some diagnostic tips.

---

## Canceling an Interactive SQL command

Use the Interrupt button on the Interactive SQL toolbar to cancel a command.

A Stop operation stops current processing and prompts for the next command. If a command file was being processed, you are prompted for an action to take (Stop Command File, Continue, or Exit Interactive SQL). You can control these actions with the Interactive SQL ON\_ERROR option.

When an interruption is detected, one of three different errors is reported, depending on when the interruption is detected. If the interruption is detected:

- When Interactive SQL is processing the request (as opposed to the database server), this message appears

```
ISQL command terminated by user
```

Interactive SQL stops processing immediately and the current database transaction is not updated.

- By the database server while processing a data manipulation command (select, insert, delete, or update), this message appears:  

```
Statement interrupted by user.
```

The effects of the current command are left unfinished, but the rest of the transaction is left intact.
- While the database server is processing a data definition command (create *object*, drop *object*, alter *object*, and so on.), this message appears:  

```
Terminated by user -- transaction rolled back
```

Since data definition commands all perform a commit automatically before the command starts, rollback simply cancels the current command.

This message also occurs when the database server is running in bulk operations mode executing a command that modifies the database (insert, update, and delete). In this case, rollback cancels not only the current command, but everything that has been done since the last commit. In some cases, it may take a considerable amount of time for the database server to perform the automatic rollback.

## Combining multiple statements

Interactive SQL allows you to enter multiple statements at the same time. End each statement with the Transact-SQL command, `go`.

- 1 Enter multiple statements in the SQL Statements pane separated by `go`:

```
update titles
set price = 21.95
where pub_id = "1389"
go
update titles
set price = price + 2.05
where pub_id = "0736"
go
update titles
set price = price+2.0
where pub_id = "0877"
go
```

- 2 On the toolbar, click the Execute SQL Statement button (or select F9). All three statements are executed. After execution, the commands remain in the SQL Statements pane. To clear this pane, press the Esc key.

You can roll back your changes by entering rollback and executing the statement.

## Looking up tables, columns, and procedures

---

**Note** You must have jConnect for JDBC installed to use the functionality described in this section.

---

Select Tools | Lookup Table Name and Tools | Lookup Procedure Name utilities to look up the names of tables, columns, or procedures stored in the current database and insert them at your cursor position.

Enter the first characters of a table, column, or procedure in the Lookup Table Name and Lookup Procedure Name dialogs. This list narrows to display only those items that start with the text you entered.

You can use the standard SQL wildcard character % to mean “match anything”. Clear the search area to display all items.

v **Looking up names of tables in the database**

- 1 Choose Tools | Lookup Table Name.
- 2 Find and select the table.
- 3 Click OK to insert the table name into the SQL Statements pane.

v **Looking up column names in the database**

- 1 Choose Tools | Lookup Table Name.
- 2 Find and select the table containing the column.
- 3 Click Show Columns.
- 4 Select the column and click OK to insert the column name into the SQL Statements pane.

v **Looking up procedure names in the database**

- 1 Choose Tools | Lookup Procedure Name.
- 2 Find and select the procedure.
- 3 Click OK to insert the procedure name into the SQL Statements pane.

## Recalling commands

When you execute a command, Interactive SQL automatically saves it in a history list that lasts for the duration of the current session. Interactive SQL maintains a record of as many as 50 of the most recent commands. You can view the entire list of commands in the Command History dialog.

To access the Command History dialog, press **Ctrl+H**, or select the book icon in the toolbar.

The most recent commands appear at the bottom of the list. To recall a command, highlight it and click **OK**. It appears in the SQL Statements pane.

You can also recall commands without the Command History dialog. Use the arrows in the toolbar to scroll back and forward through your commands, or press **Alt+Right Arrow** and **Alt+Left Arrow**.

You can save commands in text files and use them in a subsequent Interactive SQL session.

## Logging commands

With the Interactive SQL logging feature, you can record commands as you execute them. Interactive SQL continues to record until you stop the logging process, or until you end the current session. The recorded commands are stored in a log file.

### v **Beginning to log Interactive SQL commands**

- 1 Choose **SQL | Start Logging**.
- 2 In the **Save** dialog, specify a location and name for the log file.
- 3 Click **Save** when finished.

To stop logging Interactive SQL commands, select **SQL | Stop Logging**.

To start logging by typing in the SQL Statements pane, enter and execute start logging "`c:\file_name.sql`", where `c:\file_name.sql` is the path, name, and extension of the log file. A log file must have the `.sql` extension. Include the single quotation marks if the path contains embedded spaces. To stop logging, enter and execute stop logging.

---

**Note** The commands start logging and stop logging are not Transact-SQL commands, and are not supported by Adaptive Server outside the Interactive SQL dialog box.

---

Once you start logging, all commands that you try to execute are logged, including ones that do not execute properly.

## Configuring Interactive SQL

You can configure Interactive SQL in the Options dialog, which provides settings for commands, appearance, import/export features, and messages.

After you have made your selections, select either OK or Make Permanent. If you select Make Permanent, Interactive SQL starts with the configuration you selected.

You can set each option either by using the GUI, or by using the set option statement.

To access the Options dialog, choose Tools | Options.

## General dialog box

The General dialog box includes these components:

- Commit – lets you select when transactions are committed. You can commit transactions automatically after each statement is executed or only when you exit your Interactive SQL session.

You can also commit manually by entering an explicit commit command whenever appropriate. The default behavior is that transactions are committed when you exit Interactive SQL.



- Command files – determine how Interactive SQL acts when an error occurs. When an error occurs:
  - Continue – Interactive SQL displays the error message in the Results pane but does not exit. Correct the problem, then reissue the command.
  - Exit – Interactive SQL exits when an error occurs.
  - Notify and Continue – Interactive SQL displays the error message in a dialog box and describes the error but does not exit
  - Notify and Exit – Interactive SQL displays the error message in a dialog box, describes the error, and exits.
  - Notify and stop – Interactive SQL displays the error message and describes the error.
  - Prompt – the default setting. Interactive SQL displays a message box asking if you want to continue.
  - Stop – Interactive SQL displays the error message in the Results pane. Correct, then reissue the command.

Echo Command Files to Log – when you enable logging, this option causes SQL statements executed from script files (or command files) to be copied to the log along with the SQL statements entered interactively. If you disable this option, only SQL statements entered interactively are copied to the log when you start logging.

#### Folders

Determines in which directory the browser should start looking for files. Select either “Last folder used” or “Current folder.”

## Result dialog box

The Results dialog box has these components:

- Display null values – lets you specify how you want nulls to appear in the table columns when you browse data. The default setting is (NULL).
- Maximum number of rows to display – limits the number of rows that appear. The default setting is 500.
- Truncation length – limits the number of characters that appear in each column in the Results pane in Interactive SQL. The default setting is 30.

- Show multiple result sets – enables or disables the display of multiple result sets. For example, you can use this feature when you create a procedure containing multiple select statements. If this option is enabled, you can see each result set on a separate tab in the Results pane when you call the procedure.

If you are using the jConnect driver, choosing to display multiple result sets requires Interactive SQL to wait for an entire result set to be retrieved before any rows appear. This may result in longer waits for large result sets. This option is off by default.

- Show row number – check if you want the row numbers displayed in the result set.
- Automatically refetch result – enables or disables the ability of Interactive SQL to automatically regenerate the most recent result set after you execute an insert, update, or delete statement. For example, if you are updating a table with the Results tab in the Results pane displaying the rows about to be affected, this option causes Interactive SQL to automatically refresh the Results tab to reflect the executed changes. This option is on by default.
- Console Mode – select how you want the result sets displayed in the console; only the last result sets, all result sets, or no result sets.
- Font – select which font you want to use for the result set.

## Import/Export dialog box

The Import/Export dialog box has these components:

- Default export format – select the default file format for exporting. This format is automatically selected in the Files of Type field in the Save dialog, but you can still choose other formats. The default is also used when Interactive SQL interprets an output statement if no format is specified. The default setting is ASCII.
- Default import format – select the default file format for importing. This format is automatically selected in the Files of Type field in the Open dialog, but you can still choose other formats. The default is also used when Interactive SQL interprets an input statement if no format is specified. The default setting is ASCII.

- ASCII options – specify the default symbols that are used for the field separator, quote string, escape character, and the default encoding datatype when you import or export data in the ASCII format. The default settings are the comma (,) for the field separator, an apostrophe (') for the quote string, and a backslash (\) for the escape character.  
By default, Interactive SQL uses the default datatype of the server.

## Messages dialog tab

The Messages dialog box has these components:

- Measure execution time for SQL statements – enables or disables the ability of Interactive SQL to measure the time it takes for a statement to execute. When this option is enabled (which is the default), the time appears in the Messages pane.
- Show separate Messages pane – lets you specify where information from the database server appears. By default, messages appear on the Messages tab in the Results pane. If you select this option, database server information appears in a separate Messages pane in the Interactive SQL window.
- Default number of lines in Messages pane – lets you specify the initial height (in lines) of the Messages pane. The default is 7 lines.

## Editor

The Editor dialog box has these components:

- Editor – Select your scrollbar style preference: vertical, horizontal, or both.
- Tabs – This pane determines how tabs are used in your SQL text:
  - Tab size – enter the number of spaces you want each tab to comprise.
  - Indent size – Enter the number of spaces for each indent.
  - Tab radio buttons – Select Insert spaces to convert tabs to spaces when you indent SQL text. Select Keep tabs to retain tabs as spaces when you indent SQL text.
  - Auto indent – Select:

- None if you do not want to automatically indent SQL text
- Default to use the default tab and indent settings
- Smart if you want Interactive SQL to automatically indent SQL text. Select Indent open brace to indent open braces or Indent closing brace to indent the closing braces. Interactive SQL displays how these decisions affect the text in the window below the buttons.
- Format – Determines the look of your SQL text.
  - Text Highlighting – select the type of text you want to highlight from the list (keywords, comments, strings, and so on).
  - Foreground – select the foreground color of the text.
  - Background – select the color of the text's background field.
- Style
  - Font size – select the size font.
  - Caret color – determines the color of the caret.
  - Reset All – returns all styles to original selections.
- Print – Customizes your printed jobs:
  - Header – enter the header text.
  - Footer – enter the footer text.
  - Font size – select the font size.

## Query Editor

The Query Editor dialog box has these components:

- Fully qualify table and column names – select this to have Interactive SQL prefix table names with the owner (for example, `dbo.authors`) and prefix column names with the owner and table names (for example, `dbo.titles.price`).
- Quote names – Select this to automatically put quotes around table and columns names to avoid conflicts with reserved words.
- Get list of tables on startup – Select this to automatically get a list of tables in the database when the query editor is started.

## Running command files

This section describes how to process files consisting of a set of commands.

### Writing output to a file

In Interactive SQL, the output for each command remains in the Results pane until the next command is executed. To keep a record of your data, you can save the output of each statement to a separate file.

If `statement1` and `statement2` are two `select` statements, then you can output them to `file1` and `file2`, respectively, as follows:

```
Statement1; OUTPUT TO file1
go
statement2; OUTPUT TO file2
go
```

For example, the following command saves the result of a query:

```
select * from titles
go
output to "C:\My Documents\Employees.txt"
```

### Executing command files

You can execute command files by:

- Using the Interactive SQL `read` command to execute command files. The following statement executes the file `temp.sql`:

```
read temp.SQL
```

- Loading a command file into the SQL Statements pane and execute it directly from there.

You load command files back into the SQL Statements pane by choosing `File | Open`. Enter `transfer.sql` when prompted for the file name.

- Supplying a command file as a command-line argument for Interactive SQL.

The SQL Statements pane in Interactive SQL has a limit of 500 lines. For command files larger than this, you should use a generic editor capable of handling large files and use the `read` command to import it into Interactive SQL, which has no limit on the number of lines it can read.

## Saving, loading, and running command files

You can save the commands currently present in the SQL Statements pane so they are available for future Interactive SQL sessions. The file in which you save them is called a command file.

Command files are text files containing SQL statements. You can use any editor you like to create command files. You can include comment lines along with the SQL statements to be executed. These command files are commonly called scripts.

When you begin a new session, you can load the contents of a command file into the SQL Statements pane, or you can run the contents immediately.

v **Saving the commands from the SQL Statements pane to a file**

- 1 Choose File | Save.
- 2 In the Save dialog, specify a location, name, and format for the file. Click Save when finished.

v **Loading commands from a file into the SQL Statements pane**

- 1 Choose File | Open
- 2 In the Open dialog, find and select the file. Click Open when finished.

v **Running a command file immediately**

- 1 Choose File | Run Script.
- 2 The Run Script menu item is the equivalent of a read statement. For example, in the SQL Statements pane, you can also run a command file by typing the following, where *c:\filename.sql* is the path, name, and extension of the file. Single quotation marks (as shown) are required only if the path contains spaces:

```
READ "c:\filename.sql"
```

- 3 In the Open dialog, find and select the file. Click Open when finished.

The Run Script menu item is the equivalent of a READ statement. For example, in the SQL Statements pane, you can also run a command file by typing the following, where *c:\filename.sql* is the path, name, and extension of the file. Single quotation marks (as shown) are required only if the path contains spaces:

```
READ 'c:\filename.sql'
```

## Using SQL escape syntax in Interactive SQL

Interactive SQL supports JDBC escape syntax. This escape syntax allows you to call stored procedures from Interactive SQL regardless of the database management system you are using. The general form for the escape syntax is:

```
{{ keyword parameters }}
```

The braces must be doubled. This doubling is specific to Interactive SQL. There must not be a space between successive braces: "{{" is acceptable, but "{ {" is not. As well, you cannot use newline characters in the statement. The escape syntax cannot be used in stored procedures because they are not executed by Interactive SQL.

You can use the escape syntax to access a library of functions implemented by the JDBC driver that includes number, string, time, date, and system functions.

For example, to obtain the name of the current user in a database management system-neutral way, you would type the following:

```
select {{ fn user() }}
```

The functions that are available depend on the JDBC driver that you are using. The following lists the numeric functions that are supported by jConnect.

### Function names

abs	cos	log10	sign
acos	cot	pi	sin
asin	degrees	power	sqrt
atan	exp	radians	tan
atan2	floor	rand	
ceiling	log	round	

The following lists the string functions that are supported by jConnect.

### Numeric function names

ascii	difference	repeat	space
char	lcase	right	substring
concat	length	soundex	ucase

The following lists the system functions that are supported by jConnect.

### System function names

database	ifnull	user	convert
----------	--------	------	---------

The following lists the system functions that are supported by jConnect.

**System function names**

curdate	dayofweek	monthname	timestampadd
curtime	hour	now	timestampdiff
dayname	minute	quarter	year
dayofmonth	month	second	

A statement using the escape syntax should work in Adaptive Server Anywhere, Adaptive Server Enterprise, Oracle, SQL Server, or another database management system that you are connected to from Interactive SQL. For example, to obtain database properties with the sa\_db\_info procedure using SQL escape syntax, type the following in the SQL Statements pane in Interactive SQL:

```
((CANN sa_db_info(1)))
```

## Interactive SQL commands

Interactive SQL includes a set of commands that are entered in the top pane of the Interactive SQL display. These commands are intended only for Interactive SQL and are not sent to Adaptive Server for execution. Table 3-2 on page 39 lists the commands available for Interactive SQL.



**Table 3-2: DBISQL commands**

<b>Command</b>	<b>Description</b>
clear	Clears the Interactive SQL panes.
configure	Opens the Interactive SQL Options dialog.
connect	Establishes a connection to a database.
disconnect	Drops the current connection to a database.
exit	Leaves Interactive SQL.
input	Imports data into a database table from an external file or from the keyboard.
output	Imports data into a database table from an external file or from the keyboard.
parameters	Specifies parameters to an Interactive SQL command file.
read	Reads Interactive SQL statements from a file.
set connection	Changes the current database connection to another server.
set option	Use this statement to change the values of Interactive SQL options.
start logging	Use this statement to start logging executed SQL statements to a log file.
stop logging	Use this statement to stop logging of SQL statements in the current session.
system	Use this statement to launch an executable file from within Interactive SQL.

For more information about the Interactive SQL commands, see “Using DBISQL Commands” in the *Reference Manual: Commands*.



## Using *bcp* to Transfer Data to and from Adaptive Server

This chapter explains how to use the bulk copy utility, *bcp*, to move data between Adaptive Server and an operating system file.

Topic	Page
Methods for moving data	42
<i>bcp</i> requirements	43
<i>bcp</i> modes	45
Copying data to a file	45
<i>bcp</i> performance	46
Using the <i>bcp</i> options	60
Changing the defaults: interactive <i>bcp</i>	62
Using format files	72
Examples: copying out data interactively	76
Examples: copying in data interactively	79
Using <i>bcp</i> with alternate languages	83
<i>bcp</i> and row-level access rules	84
Copy in and batch files	84
Copy out and text and image data	86
Specifying a network packet size	87
Copy in and error files	87
Copy out and error files	88
Data integrity: defaults, rules, and triggers	89
How <i>bcp</i> differs from other utilities	89

*bcp* provides a convenient, high-speed method for transferring data between a database table or view and an operating system file. *bcp* can read or write files in a wide variety of formats. When copying in from a file, *bcp* inserts data into an existing database table; when copying out to a file, *bcp* overwrites any previous contents of the file.

For a detailed description of *bcp* syntax, see *bcp* on page 128.

## Methods for moving data

You can use the following methods to move data to and from your Adaptive Server databases:

- `bcp` as a standalone program from the operating system. This chapter provides instructions for this method.
- Client-Library, which calls bulk library routines. For more information about the Client-Library, see the *Open Client and Open Server Common Libraries Reference Manual*.

## Importing and exporting data with `bcp`

Transact-SQL commands cannot transfer data in bulk. For this reason, you must use `bcp` for any large transfers. You can use `bcp` to:

- Import data that was previously associated with another program, such as the records from another database management system. This is the most common use for `bcp`.

Before using `bcp`, you must create a file of the records you want to import. The general steps are:

- a Put the data to transfer into an operating system file.
  - b Run `bcp` from the operating system command line.
- Move tables between Adaptive Servers or between Adaptive Server and other data sources that can produce an operating-system file.
  - Copy out data from a view. See `bcp` on page 128 for a description of the syntax for using `bcp` to copy out from a view.

---

**Note** You cannot use `bcp` to copy **in** data to a view.

---

- Transfer data for use with other programs, for example, with a spreadsheet program. The general steps to transfer data are:
  - a Use `bcp` to move the data from Adaptive Server into an operating-system file from which the other program imports the data.
  - b When you finish using your data with the other program, copy it into an operating-system file, and then use `bcp` to copy it into Adaptive Server.

Adaptive Server can accept data in any character or binary format, as long as the data file describes either the length of the fields or the **terminators**, the characters that separate columns.

The structures in the tables involved in the transfer need not be identical, because when *bcp*:

- Imports **from** a file, it appends data to an existing database table.
- Exports **to** a file, it overwrites the previous contents of the file.

When the transfer is complete, *bcp* informs you of the:

- Number of rows of data successfully copied
- Number of rows (if any) that it could not copy
- Total time the copy took
- Average amount of time, in milliseconds, that it took to copy one row
- Number of rows copied per second.

If *bcp* runs successfully, you see a return status of 0. The return status generally reflects errors from the operating system level and correspond to the ones listed in the *errno.h* file in the */usr/include/sys/* directory.

## ***bcp* requirements**

Before using *bcp*, you need to provide it with basic data information and prepare both the data for transfer and the command to access the data.

### Basic requirements

You must supply the following information to transfer data successfully to and from Adaptive Server:

- Name of the database and table or view
- Name of the operating system file
- Direction of the transfer (in or out)

You can also use *bcp* to modify the storage type, storage length, and terminator for each column if you want to do so.

### Permissions

You must have an Adaptive Server account and the appropriate permissions on the database tables or views, as well as the operating system files to use in the transfer to use *bcp*.

- To copy data into a table, you must have insert and select permission on the table.
- To copy a table to an operating system file, you must have select permission on the following tables:
  - the table to copy
  - sysobjects
  - syscolumns
  - sysindexes

Pre-transfer tasks

Before you can use bcp in, you must prepare the command and the data for transfer:

- To use either fast or slow bcp, set select into/bulkcopy/pllsort to true. For example, to turn on this option for the pubs2 database, you would enter:

```
sp_dboption pubs2, "select into/bulkcopy/pllsort", true
```

For more information, see “bcp modes” on page 45.

- To use fast bcp, remove indexes (except non-clustered indexes) on the target table. For more information about this requirement, see “bcp performance” on page 46.

In addition:

- If you are running Open Client version 11.1 or later and are using an external Sybase configuration file, you must add the following to enable bcp:

```
[BCP]
```

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use bcp.
- To use a previous version of bcp, you must set the CS\_BEHAVIOR property in the [bcp] section of the *ocs.cfg* file:

```
[bcp]  
CS_BEHAVIOR = CS_BEHAVIOR_100
```

If CS\_BEHAVIOR is not set to CS\_BEHAVIOR\_100, you can use functionality for bcp 11.1 and later.

## ***bcp* modes**

bcp in works in one of two modes:

- Slow bcp – logs each row insert that it makes, used for tables that have one or more indexes.
- Fast bcp – logs only page allocations, copying data into tables without indexes or at the fastest speed possible. You can use fast bcp on tables with nonclustered indexes.

To determine the bcp mode that is best for your copying task, consider the:

- Size of the table into which you are copying data
- Amount of data that you are copying in
- Number of indexes on the table
- Amount of spare database device space that you have for re-creating indexes

---

**Note** Fast bcp might enhance performance; however, slow bcp gives you greater data recoverability.

---

## **Copying data to a file**

You can use bcp to copy data from a table to an operating system file. You can specify a table name and one or more partitions as the source. Data can be copied:

- To a single file for all partitions
- To a single file for each partition

If you do not specify a destination file name, Adaptive Server creates file names based on the partition names.

## **Examples**

In these examples, we are copying data from bigtable, which is partitioned three ways: ptn1, ptn2, and ptn3, to various operating system files.

To copy the data in bigtable to *file1*, enter:

```
bcp mydb..bigtable out file1
```

To copy the data from ptn1, ptn2, and ptn3 to *file2*, enter:

```
bcp mydb..bigtable partition ptn1, ptn2, ptn3 out file2
```

To copy the data from ptn1 and ptn2 to data files *ptn1.dat* and *ptn2.dat*, enter:

```
bcp mydb..bigtable partition ptn1, ptn2 out ptn1.dat,  
ptn2.dat
```

This example also copies the data from ptn1 and ptn2 to *ptn1.dat* and *ptn2.dat*.

Enter:

```
bcp mydb..bigtable partition ptn1, ptn2 out
```

See Chapter 7, “Utility Commands Reference,” for complete syntax and usage information for bcp.

## ***bcp performance***

Keeping indexes on a table causes the bulk copy utility to use slow bcp automatically. However, slow bcp can fill the transaction log very quickly.

- When you are copying a large number of rows, the performance penalty and log space requirements for using slow bcp can be severe.
- For extremely large tables, using slow bcp is not an option because its detailed log makes it much too slow.

To improve the performance of bcp:

- Use partitioned tables. Several bcp sessions with a partitioned table can reduce dramatically the time required to copy the data. However, such performance improvements are more noticeable in fast bcp than in slow bcp.
- Use bcp in parallel to increase performance dramatically. Parallel bulk copy can provide balanced data distribution across partitions. For more information, see “Using parallel bulk copy to copy data into a specific partition” on page 53.



## Using fast or slow *bcp*

The existence of indexes on tables affects transfer speed. When you use *bcp* on such tables, *bcp* automatically uses its slow mode, which logs data inserts in the transaction log. These logged inserts can cause the transaction log to become very large.

To control this data excess and ensure that the database is fully recoverable in the event of a failure, you can back up the log with *dump transaction*.

---

**Note** *bcp* does not fire any trigger that exists on the target table.

---

Fast *bcp* logs only the page allocations. For copying data in, *bcp* is fastest if your database table has no indexes.

However, if you used fast *bcp* to make data inserts, which fast *bcp* does not log, you cannot back up (*dump*) the transaction log to a device. The changes are not in the log, and a restore cannot recover nonexistent backup data. The requested backup (*dump transaction*) produces an error message that instructs you to use *dump database* instead. This restriction remains in force until a *dump database* successfully completes. For more information about *dump database* and *dump transaction*, see the *System Administration Guide*, and the *Reference Manual*.

When does Adaptive Server use slow *bcp*?

- `sp_dboption 'select into/bulkcopy/pllsort'` is off.
- `sp_dboption 'select into/bulkcopy/pllsort'` is on, but the table uses the `allpages` locking scheme and has a clustered index.
- `sp_dboption 'select into/bulkcopy/pllsort'` is on, but the table has a unique nonclustered index.

If the option `ignore_dup_key` option is enabled on the unique index, performing fast *bcp* can put the table and index in an inconsistent state if rows with duplicate keys are inserted. To avoid the inconsistency, Adaptive Server performs slow *bcp*.

- If the table has nonclustered indexes or triggers, and the table is marked for replication or the database is used as a warm standby.

Because fast *bcp* does not log inserts, if Adaptive Server uses fast *bcp*, the rows *bcp* copies cannot be recovered on the replication site if there is a problem. Adaptive Server uses slow *bcp* in these situations to maintain compatibility with applications that were written to use the old behavior.

When does Adaptive Server use fast *bcp*?

These are situations in which Adaptive Server uses fast *bcp* (in all cases `sp_dboption 'select into/bulkcopy/pllsort'` is enabled and the table does not have a clustered index):

- When the table has a non-unique, nonclustered index. Adaptive Server logs the index updates and the page allocations only. It does not log inserts into the table.
- When a table has triggers. However, bcp does not fire any triggers in the target table.
- When a table has datarows or datapage locking scheme with a clustered index.

If the table includes nonclustered indexes or triggers, but `sp_dboption 'select into/bulkcopy/pllsort'` is not enabled, Adaptive Server uses slow bcp, and prints this warning message to indicate that you can improve the performance of bcp by enabling `sp_dboption 'select into/bulkcopy/pllsort'`:

```
Performing slow bcp on table '%s'. To enable fast bcp
please turn on 'select into/bulkcopy' option on the
database '%s
```

bcp optimization is performed by Adaptive Server and does not require that you use Open Client version 15.0 or later.

#### Copying tables with indexes

The bcp program is optimized to load data into tables that do not have indexes associated with them. It loads data into tables without indexes at the fastest possible speed, with a minimum of logging. Page allocations are logged, but the insertion of rows is not.

When you copy data into a table that has one or more indexes, you can use fast bcp. This includes indexes implicitly created using the unique integrity constraint of a create table statement. However, bcp does not enforce the other integrity constraints defined for a table.

By default, the `select into/bulkcopy/pllsort` option is false (off) in newly created databases. To change the default setting for future databases, turn this option on in the model database.

---

**Note** The log can grow very large during slow bcp because bcp logs inserts into a table that has indexes. After the bulk copy completes, back up your database with `dump database`, then truncate the log with `dump transaction` after the bulk copy completes and after you have backed up your database with `dump database`.

---

While the `select into/bulkcopy/plsort` option is on, you cannot dump the transaction log. Issuing dump transaction produces an error message instructing you to use `dump database` instead.

---

**Warning!** Be certain that you dump your database before you turn off the `select into/bulkcopy/plsort` flag. If you have inserted unlogged data into your database, and you then perform a dump transaction before performing a dump database, you will not be able to recover your data.

---

Fast bcp runs more slowly while a dump database is taking place.

Table 4-1 shows which version bcp uses when copying in, the necessary settings for the `select into/bulkcopy/plsort` option, and whether the transaction log is kept and can be dumped.

**Table 4-1: Comparing fast and slow bcp**

<code>select into/bulkcopy/plsort</code>	on	off
fast bcp ( <i>no cluster indexes on target table</i> )	OK dump transaction prohibited	bcp prohibited dump transaction
slow bcp (one or more <i>cluster indexes</i> )	OK dump transaction prohibited	OK dump transaction OK

---

**Note** The performance penalty for copying data into a table that has indexes in place can be severe. If you are copying in a very large number of rows, it may be faster to drop all the indexes beforehand with `drop index` (or `alter table`, for indexes created as a unique constraint); set the database option; copy the data into the table; re-create the indexes; and then dump the database. Remember to allocate disk space for the construction of indexes: about 2.2 times the amount of space needed for the data.

---

#### Configuring databases for fast bcp

To allow a user to copy in data using fast bcp, either a System Administrator or the Database Owner first must use `sp_dboption` to set `select into/bulkcopy/plsort` to true on the database that contains the target table or tables. If the option is set to false when a user tries to use fast bcp to copy data into a table without indexes, Adaptive Server generates an error message.

---

**Note** You do not need to set the `select into/bulkcopy/plsort` option to true to copy out data from, or to copy in data to a table that has indexes. Slow bcp always copies tables with indexes and logs all inserts.

---

By default, the `select into/bulkcopy/pllsort` option is set to `false` (off) in newly created databases. To change the default setting for future databases, turn this option on (set to `true`) in the model database.

### Dropping indexes

If you are copying a very large number of rows, you must have 1.2 times the amount of space needed for the data and enough space for the server to reconstruct a clustered index.

- If space is available, you can use `drop index` to drop all the indexes beforehand.
- If you do not have enough space for the server to sort the data and build the index or indexes, use `slow bcp`.

## Copying in data with fast bcp

Table 4-2 summarizes the steps for copying in data to Adaptive Server using `fast bcp`.

**Table 4-2: Steps for copying in data using fast bcp**

Step	Who can do it
Use <code>sp_dboption</code> to set <code>select into/bulkcopy/pllsort</code> to <code>true</code> . Run checkpoint in the database that was changed.	System Administrator or Database Owner
Have enough space to re-create any indexes on the table. Drop the indexes on the table.	Table owner
Have insert permission on the table.	Granted by the table owner
Perform the copy with <code>bcp</code> .	Any user with insert permission
Re-create the indexes.	Table owner
Reset <code>sp_dboption</code> , if desired, and run checkpoint in the database that was changed.	System Administrator or Database Owner
Use <code>dump database</code> to back up the newly inserted data.	System Administrator, Operator, or Database Owner
Run stored procedures or queries to determine whether any of the newly loaded data violates rules.	Table owner or stored procedure owner

## Bulk copying data into partitioned tables

In certain circumstances, you can improve `bcp` performance dramatically by executing several `bcp` sessions with a partitioned table.

Partitioned tables improve insert performance by reducing lock contention and by distributing I/O over multiple devices. bcp performance with partitioned tables is improved primarily because of this distributed I/O.

When you execute a bcp session on a partitioned table, consider:

- A partitioned table improves performance when you are bulk copying **in** to the table.
- The performance of slow bcp does not improve as much with partitioned tables. Instead, drop all indexes and use fast bcp, as described in Table 4-2 on page 50, to increase performance.
- Network traffic can quickly become a bottleneck when multiple bcp sessions are being executed. If possible, use a local connection to the Adaptive Server to avoid this bottleneck.

When copying data into a partitioned table, you can:

- Copy the data randomly without regard to the partition to which data is copied. For example, to copy data from *file1* to *bigtable*, enter:

```
bcp mydb..bigtable in file1
```

To copy data from *file1*, *file2*, and *file3* to *bigtable*, enter:

```
bcp mydb..bigtable in file1, file2, file3
```

- Copy the data into a specific partition. For example, to copy data from *file1* to *ptn1*, *file2* to *ptn2*, and *file3* to *ptn3*, enter:

```
bcp mydb..bigtable partition ptn1, ptn2, ptn3 in
file1, file2, file3
```

To copy data from *file1* to the first partition of *bigtable*, enter:

```
bcp mydb..bigtable:1 in file1
```

If the table has a clustered index, bcp runs in slow mode and allows the index to control the placement of rows.

See Chapter 7, “Utility Commands Reference,” for complete syntax and usage information for bcp.

#### v **Copying data randomly into partitions**

To copy data randomly into partitioned tables when using multiple bcp sessions, you must:

- 1 Configure the table with as many partitions and physical devices as you require for your system.

For more information, see the *Performance and Tuning Guide*, and “Using parallel bulk copy to copy data into a specific partition” on page 53 of this manual.

- 2 Make sure Adaptive Server is configured with enough locks to support multiple bcp sessions. For information on configuring locks, see the *System Administration Guide*.
- 3 Remove the indexes on the table and enable fast bcp. See “Using fast or slow bcp” on page 47 for instructions.

---

**Note** If you use slow bcp, performance may improve significantly after you remove the indexes.

---

- 4 Divide the bcp input file into as many files of equal size as the number of planned simultaneous bcp sessions.

You also can use the `-F first_row` and `-L last_row` options to specify the start and end of each “input file.”

- 5 Execute the bcp sessions with separate files in parallel on the local Adaptive Server machine.

For example, on UNIX platforms, you can execute different sessions in different shell windows or start individual bcp sessions in the background.

Read the *Performance and Tuning Guide* for a detailed description of copying data into partitioned tables.

## Monitoring bcp sessions with dbcc checktable and sp\_helpsegment

If you do not specify which partition the bcp sessions should use, Adaptive Server randomly assigns the multiple bcp sessions to the table’s available partitions. If this random assignment occurs, be sure to monitor the partitions to ensure that the process has evenly distributed the inserts by using either of the following:

- `dbcc checktable` – to periodically to check the total page counts for each partition
- `sp_helpsegment` or `sp_helppartition` – to perform a similar check, but without locking the database objects

For more information about `dbcc checktable`, see the *System Administration Guide*. For more information about `sp_helpsegment` and `sp_helppartition`, see the *Reference Manual*.

For more information about table partitions, see the *Performance and Tuning Guide*.

## Reducing logging by increasing page allocations

If you are using fast bcp, consider that each bcp in batch requires the page manager to allocate one or more extents. Each such allocation generates a single log record.

Use the number of preallocated extents configuration parameter to specify how many extents Adaptive Server is to allocate through the page manager.

- Valid values for the number of preallocated extents configuration parameter are from 0 to 31; the default value is 2.
- You must restart Adaptive Server to change the value.
- When performing large bcp operations, increase this number to prevent the page allocations from filling the log.
- Set this value to 0 to prevent large extent allocations, so that the page manager performs only single-page allocations.

Adaptive Server may allocate more pages than are actually needed, so keep the value small when space is limited. These pages are deallocated at the end of the batch.

For more information, see the *System Administration Guide*.

## Using parallel bulk copy to copy data into a specific partition

Use parallel bulk copy to copy data in parallel to a specific partition. Parallel bulk copy substantially increases performance during bcp sessions because it can split large bulk copy jobs into multiple sessions and run the sessions concurrently.

To use parallel bulk copy:

- The destination table must be partitioned.
  - Use `sp_helppartition` to see the number of partitions on the table.
  - Use `alter table ... partition` to partition the table, if the table is not already partitioned.
- The destination table should not contain indexes because:

- If the table has a clustered index, this index determines the physical placement of the data, causing the partition specification in the bcp command to be ignored.
- If any indexes exist, bcp automatically uses its slow bulk copy instead of its fast bulk copy mode.
- If nonclustered indexes exist on the tables, parallel bulk copy is likely to lead to deadlocks on index pages.
- Each partition should reside on a separate physical disk for the best performance.
- Before you copy data into your database, you must partition the table destined to contain the data.
- Parallel bulk copy can copy in to a table from multiple operating system files.

For all types of partitioned tables, use:

`bcp tablename partition partition_name in file_name`

For round-robin partitioned tables only, you can use:

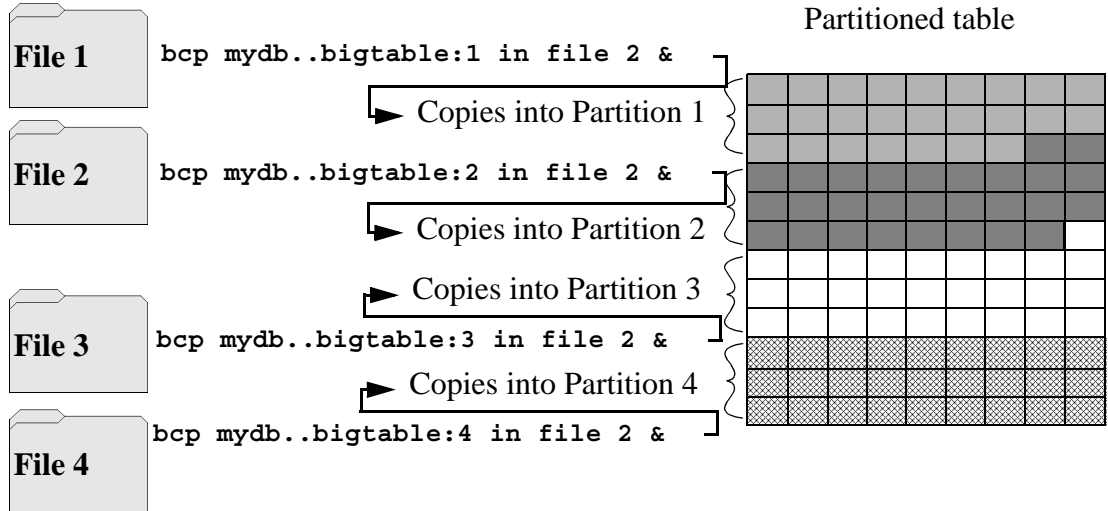
`bcp tablename partition_number in file_name`

Figure 4-1 illustrates the parallel bulk copy process.



**Figure 4-1: Copying data into a round-robin partitioned table using parallel bulk copy**

Large file divided  
into four smaller files



See the *Performance and Tuning Guide* for information about partitioning a table.

---

**Note** When using parallel bulk copy to copy data out, you cannot specify a partition number. You can specify a partition name.

---

## **bcp in and locks**

When you copy in to a table using bcp, and particularly when you copy in to a table using parallel bcp, the copy process acquires the following locks:

- An exclusive intent lock on the table
- An exclusive page lock on each data page or data row
- An exclusive lock on index pages, if any indexes exist

If you are copying in very large tables, and especially if you are using simultaneous copies into a partitioned table, this can involve a very large number of locks.

To avoid running out of locks:

- Increase the number of locks.
  - To estimate the number of locks needed, use:

*# of simultaneous batches \* (rows\_per\_batch / (2016/row\_length))*

- To see the row length for a table, use:

```
1> select maxlen
2> from sysindexes
3> where id = object_id("tablename") and (indid = 0 or indid = 1)
```

See the *System Administration Guide* for more information about setting the number of locks.

- Use the `-b batchsize` flag to copy smaller batches; the default batch size is 1000 rows.
- Run fewer batches concurrently.

## Parallel bulk copy methods

Use one of the following methods to copy in data using parallel bulk copy:

- Start multiple bcp sessions in the background, being sure to:
  - Specify the password at the command line.
  - Use native mode, character mode, or a format file.You can start bcp as many times as the table is partitioned.
- Create and use a format file:
  - a Start bcp in interactive mode.
  - b Answer the prompts.
  - c Create a format file that stores your responses.
  - d Put the process in the background when the copy begins.
  - e Issue the next bcp command, and specify the format file created with the first bcp command.
- Start bcp sessions in multiple windows.
- Specify a partition to file mapping in a single bcp in command.

The client can execute independent bcp in sessions in parallel. Alternatively, the user can specify the `--maxconn` option to control the maximum number of parallel connections that the bcp client can open to the server.

## Parallel bulk copy syntax

The syntax for parallel bulk copy is:

```
bcp table_name[:partition_number | partrition_name] in file_name
-Pmypassword
```

Where:

- *table\_name* – is the name of the table into which you are copying the data
- *partition\_name* – is the name of the partition into which you are copying
- *file\_name* – is the host file that contains the data
- *mypassword* – is your password

## Using parallel bulk copy on round-robin partitioned tables

To copy sorted data in parallel into a specific partition:

- Specify the partition by appending a colon (:) plus the partition number to the table name. For example:

```
publishers:10
```

---

**Note** The partition you specify must exist before you issue the bcp command.

---

- Split the sorted data into separate files, or delineate the “files” by specifying the first row (-F *first\_row*) and the last row (-L *last\_row*) of the host file.
- Note the number of partitions in the table. This number limits the number of parallel bulk copy sessions that you can start.

For example, if a table has four partitions, and you start five parallel bulk copy jobs, only the first four jobs can run in parallel; the fifth job does not start until one of the first four jobs finish.

bcp copies each file or set of line numbers to a separate partition. For example, to use parallel bulk copy to copy in sorted data to mydb..bigtable from four files into four partitions, enter:

```
bcp mydb..bigtable:1 in file1 -Pmypassword -c &
bcp mydb..bigtable:2 in file2 -Pmypassword -c &
bcp mydb..bigtable:3 in file3 -Pmypassword -c &
bcp mydb..bigtable:4 in file4 -Pmypassword -c &
```

## Parallel bulk copy and IDENTITY columns

When you are using parallel bulk copy, IDENTITY columns can cause a bottleneck. As bcp reads in the data, the utility both generates the values of the IDENTITY column and updates the IDENTITY column's maximum value for each row. This extra work may adversely affect the performance improvement that you expected to receive from using parallel bulk copy.

To avoid this bottleneck, you can explicitly specify the IDENTITY starting point for each session.

### Retaining sort order

If you copy sorted data into the table without explicitly specifying the IDENTITY starting point, bcp might not generate the IDENTITY column values in sorted order. Parallel bulk copy reads the information into all the partitions simultaneously and updates the values of the IDENTITY column as it reads in the data.

A bcp statement with no explicit starting point would produce IDENTITY column numbers similar to those shown in Figure 4-2:

**Figure 4-2: Producing IDENTITY columns in sorted order**

Partition 1			Partition 2			Partition 3			Partition 4		
ID column			ID column			ID column			ID column		
100	A		102	C		103	F		101	H	
104	A		106	C		105	F		110	H	
107	B		109	C		111	F		113	I	
108	B		112	D		116	G		115	J	
114	B		117	E		119	G		118	J	

The table has a maximum IDENTITY column number of 119, but the order is no longer meaningful.

If you want Adaptive Server to enforce unique IDENTITY column values, you must run bcp with either the -g or -E parameter.

### Specifying the starting point from the command line

Use the -g *id\_start\_value* flag to specify an IDENTITY starting point for a session in the command line.

The `-g` parameter instructs Adaptive Server to generate a sequence of `IDENTITY` column values for the `bcp` session without checking and updating the maximum value of the table's `IDENTITY` column for each row. Instead of checking, Adaptive Server updates the maximum value at the end of each batch.

---

**Warning!** Be cautious about creating duplicate identity values inadvertently when you specify identity value ranges that overlap.

---

To specify a starting `IDENTITY` value, enter:

```
bcp [-gid_start_value]
```

For example, to copy in four files, each of which has 100 rows, enter:

```
bcp mydb..bigtable in file1 -g100
bcp mydb..bigtable in file2 -g200
bcp mydb..bigtable in file3 -g300
bcp mydb..bigtable in file4 -g400
```

Using the `-g` parameter does not guarantee that the `IDENTITY` column values are unique. To ensure uniqueness, you must:

- Know how many rows are in the input files and what the highest existing value is. Use this information to set the starting values with the `-g` parameter and generate ranges that do not overlap.

In the example above, if any file contains more than 100 rows, the identity values overlap into the next 100 rows of data, creating duplicate identity values.

- Make sure that no one else is inserting data that can produce conflicting `IDENTITY` values.

### Specifying the starting point using the data file

Use the `-E` parameter to set the `IDENTITY` starting point explicitly from the data file.

The `-E` parameter instructs `bcp` to prompt you to enter an explicit `IDENTITY` column value for each row. If the number of inserted rows exceeds the maximum possible `IDENTITY` column value, Adaptive Server returns an error.

## Using the bcp options

The information in this section clarifies some of the more complex options of the bcp syntax. For a complete description of the syntax, see bcp on page 128.

### Using the default formats

bcp provides two command-line options that create files with frequently used default formats. These options provide the easiest way to copy data in and out from Adaptive Server.

- The `-n` option uses “native” (operating system) formats.
- The `-c` option uses “character” (char datatype) for all columns. This datatype supplies tabs between fields on a row and a newline terminator, such as a carriage return, at the end of each row.

When you use the native or character options, bcp operates noninteractively and only asks you for your Adaptive Server password.

### Native format

The `-n` option creates files using **native** (operating system-specific) formats. Native formats usually create a more compact operating system file. For example, the following command copies the publishers table to the file called `pub_out`, using native data format:

```
bcp pubs2..publishers out pub_out -n
```

Here are the contents of `pub_out`:

```
0736^MNew Age Books^FBoston^BMA0877^PBinnet & Hardley^J  
Washington^BDC1389^TAlgodata Infosystems^HBerkeley^BCA
```

bcp prefixed each field, except the `pub_id`, which is a `char(4)` datatype, with an ASCII character equivalent to the length of the data in the field. For example, “New Age Books” is 13 characters long, and `^M` (Ctrl-m) is ASCII 13.

All the table data stored in the `pub_out` file is `char` or `varchar` data, so it is human-readable. In a table with numeric data, bcp writes the information to the file in the operating system’s data representation format, which may not be human-readable.

bcp can copy data out to a file either as its native (database) datatype or as any datatype for which implicit conversion is supported for the datatype in question. bcp copies user-defined datatypes as their base datatype or as any datatype for which implicit conversion is supported. For more information on datatype conversions, see dbconvert in the *Open Client DB-Library/C Reference Manual* or the *Adaptive Server Enterprise Reference Manual*.

---

**Note** The bcp utility does not support copying data in native format from different operating systems; for example, copying from Windows to UNIX. Use the -c flag if you need to use bcp to copy files from one operating system to another.

---

---

**Warning!** Do not use row terminator (-t) or field terminator (-r) parameters with bcp in native format. Results are unpredictable and data may be corrupted.

---

## Character format

Character format (-c) uses the char datatype for all columns. It inserts tabs between fields in each row and a newline terminator at the end of each row.

For example, the following command copies out the data from the publishers table in character format to the file *pub\_out*:

```
bcp pubs2..publishers out pub_out -c
```

The command produces the following bcp output:

0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

## Changing terminators from the command line

Terminators are the characters that separate data fields (field terminators). The row terminator is the field terminator of the last field in the table or file. Use the *-tfield\_terminator* and *-rrow\_terminator* command line options with the character format option (-c) to change the terminators from the command line.

The following example uses the comma (,) as the field terminator and return (\r) as the row terminator.

- In UNIX platforms:

```
bcp pubs2..publishers out pub_out -c -t , -r \\r
```

Remember to “escape” the backslash, if necessary, for your operating system command shell.

- In Windows:

```
bcp pubs2..publishers out pub_out -c -t , -r \r
```

This *bcp* command line produces the following information:

```
0736,New Age Books,Boston,MA
0877,Binnet & Hardley,Washington,DC
1389,Algodata Infosystems,Berkeley,CA
```

---

**Note** You can use the *-t* and *-r* options to change the default terminators without including the character option (*-c*).

---

## Changing the defaults: interactive *bcp*

If you do not specify native (*-n*) or character (*-c*) format, *bcp* prompts you interactively for:

- The file storage type
- The prefix length
- The terminator for each column of data to be copied
- A field length for fields that are to be stored as char or binary

The default values for these prompts produce the same results as using the native format and provide a simple means for copying data out of a database for later reloading into Adaptive Server.

If you are copying data to or from Adaptive Server for use with other programs, base your answers to the prompts on the format required by the other software.

These four prompts provide an extremely flexible system that allows you either to read a file from other software or to create a file that requires little or no editing to conform to many other data formats.

The following sections discuss these prompts and the way they interact to affect the data.



## Responding to *bcp* prompts

When you copy data in or out using the *-n* (native format) or *-c* (character format) parameters, *bcp* prompts you only for your password, unless you supplied it with the *-P* parameter. If you do not supply either the *-n*, *-c* or *-f formatfile* parameter, *bcp* prompts you for information for each field in the table or view.

- Each prompt displays a default value, in brackets, which you can accept by pressing Return. The prompts include:
  - The file storage type, which can be character or any valid Adaptive Server datatype
  - The prefix length, which is an integer indicating the length in bytes of the following data
  - The storage length of the data in the file for non-NULL fields
  - The field terminator, which can be any character string
  - **Windows** – Scale and precision for numeric and decimal data types

The row terminator is the field terminator of the last field in the table, view, or file.

- The bracketed defaults represent reasonable values for the datatypes of the field in question. For the most efficient use of space when copying out to a file:
  - Use the default prompts
  - Copy all data in the datatypes defined by their table
  - Use prefixes as indicated
  - Do not use terminators
  - Accept the default lengths

Table 4-3 shows the *bcp* prompts, defaults, and the possible alternate user responses:

**Table 4-3: Defaults and user responses for *bcp* prompts**

Prompt	Default provided	Possible user response
File Storage Type	Use database storage type for most fields except: <ul style="list-style-type: none"> <li>• char for varchar</li> <li>• binary for varbinary</li> </ul>	char to create or read a human-readable file; any Adaptive Server datatype where implicit conversion is supported.

Prompt	Default provided	Possible user response
Prefix Length	<ul style="list-style-type: none"> <li>• 0 for fields defined with char datatype (not storage type) and all fixed-length datatypes</li> <li>• 1 for most other datatypes</li> <li>• 2 for binary and varbinary saved as char</li> <li>• 4 for text and image</li> </ul>	0 if no prefix is desired; otherwise, defaults are recommended.
Storage Length	<ul style="list-style-type: none"> <li>• For char and varchar, use defined length.</li> <li>• For binary and varbinary saved as char, use double the defined length.</li> <li>• For all other datatypes, use maximum length needed to avoid truncation or data overflow.</li> </ul>	Default values, or greater, are recommended.
Field or Row Terminator	None	Up to 30 characters, or one of the following: <ul style="list-style-type: none"> <li>• \t – tab</li> <li>• \n – newline</li> <li>• \r – carriage return</li> <li>• \0 – null terminator</li> <li>• \ – backslash</li> </ul>

## File storage type

The file storage type prompt offers you choices about how to store the data in the file. You can copy data into a file as:

- Its database table type,
- A character string, or
- Any datatype for which implicit conversion is supported.

---

**Note** bcp copies user-defined datatypes as their base types.

---

Table 4-4 shows the default storage type for each Adaptive Server datatype and the abbreviations that are acceptable to bcp.

- For the most compact storage, use the default value.
- For character files, use char.
- Keep in mind that the date storage type is the Adaptive Server internal storage format of datetime, not the host operating system format of the date.

- timestamp data is treated as binary(8).

In Table 4-4, brackets [ ] indicate that you can use the initial character or the beginning characters of the word. For example, for “bit” you can use “b,” “bi,” or “bit.”

**Table 4-4: File storage datatypes for bcp**

Table datatype	Storage type
char, varchar	c[har]
text	T[ext]
int	i[nt]
smallint	s[mallint]
tinyint	t[inyint]
float	f[loat]
money	m[oney]
bit	b[it]
datetime	d[atetime]
binary, varbinary, timestamp	x
image	I[mage]
smalldatetime	D
real	r
smallmoney	M
numeric	n
decimal	e

To display this list while using bcp interactively, type a question mark (?) in response to the prompt “Enter the file storage type”.

The suggested values that appear in the prompts are the defaults. Remember that your response determines how the data is stored in the output file; you need not indicate the column’s type in the database table.

bcp fails if you enter a type that is not either implicitly convertible or char. For example, you may not be able to use smallint for int data (you may get overflow errors), but you can use int for smallint.

When storing noncharacter datatypes as their database types, bcp writes the data to the file in Adaptive Server’s internal data representation format for the host operating system, rather than in human-readable form.

Before copying data that is in character format from a file into a database table, check the datatype entry rules in the *Reference Manual*. Character data copied into the database with bcp must conform to those rules. Note especially that dates in the undelimited (yy)ymmdd format may result in overflow errors if the year is not specified first.

When you send host data files to sites that use terminals different from your own, inform them of the *datafile\_charset* that you used to create the files.

## Prefix length

By default, bcp precedes each field that has a variable storage length with a string of one or more bytes indicating the length of the field. This prefix enables the most compact file storage.

The default values in the prompts indicate the most efficient prefix length:

- For fixed-length fields, the prefix length should be 0.
- For fields of 255 bytes or less, the default prefix length is 1.
- For text or image datatypes, the default prefix length is 4.
- For binary and varbinary datatypes that are being converted to char storage types, the default prefix length is 2, since each byte of table data requires 2 bytes of file storage.
- For binary, varbinary, and image data, use even numbers for the prefix and length. This requirement maintains consistency with Adaptive Server, which stores data as an even number of hexadecimal digits.
- For any data column that permits null values, use a prefix length, other than 0, or a terminator to denote the length of each row's data. bcp considers such columns, including columns with integer datatypes that might ordinarily be considered fixed-length columns, to be of variable length.
- For data with no prefix before its column, use a prefix length of 0.

A prefix length is a 1-, 2-, or 4-byte integer that represents the length of each data value in bytes. It immediately precedes the data value in the host file.

Unless you supply a terminator, bcp pads each stored field with spaces to the full length specified at the next prompt, "length."

Because prefix lengths consist of **native** format integers, the resulting host file contains nonprintable characters. The nature of these characters could prevent you from printing the host file or from transmitting it through a communications program that cannot handle non-human-readable characters.

For more information about prefix lengths, see Table 4-9 on page 76.

## Field length

In almost all cases, use the bcp default value for the storage length while copying data out.

---

**Note** The terms “length” and “storage length” in this section refer to the operating system file, not to Adaptive Server field lengths.

---

- If you are creating a file to reload into Adaptive Server, the default prefixes and length keep the storage space needed to a minimum.
- If you are creating a human-readable file, the default length prevents the truncation of data or the creation of overflow errors that cause bcp to fail.

Because you can change the default length by supplying another value, you must be familiar with the data to transfer. If you are copying character data in from other software, examine the source file carefully before choosing length values.

---

**Note** If the storage type is noncharacter, bcp stores the data in the operating system’s native data representation and does not prompt for a length.

---

When bcp converts noncharacter data to character storage, it suggests a default field length that is large enough to store the data without truncating datetime data or causing an overflow of numeric data.

- The default lengths are the number of bytes needed to display the longest value for the Adaptive Server datatype. Table 4-5 lists the default field lengths for data conversion to character storage.

**Table 4-5: Default field lengths for noncharacter to character datatypes**

Datatype	Default size
int	12 bytes
smallint	6 bytes
tinyint	3 bytes

Datatype	Default size
float	25 bytes
money	24 bytes
bit	1 byte
datetime	26 bytes
smalldatetime	26 bytes
real	25 bytes
smallmoney	24 bytes

- If you specify a field length that is too short for numeric data when copying data out, bcp prints an overflow message and does not copy the data.
- The default length for binary and varbinary fields is twice the length defined for the column, since each byte of the field requires 2 bytes of file storage.
- If you accept the default storage length, the actual amount of storage space allocated depends on whether or not you specify a prefix length and terminators.
  - If you specify a prefix length of 1, 2, or 4, bcp uses a storage space of the actual length of the data, plus the length of the prefix, plus any terminators.
  - If you specify a prefix length of 0 and no terminator, bcp allocates the maximum amount of space shown in the prompt, which is the maximum space that may be needed for the datatype in question. In other words, bcp treats the field as if it were fixed length to determine where one field ends and the next begins.

For example, if the field is defined as varchar(30), bcp uses 30 bytes for each value, even if some of the values are only 1 character long.

- Fields defined in the database as char, nchar, and binary, and those that do not permit null values, are always padded with spaces (null bytes for binary) to the full length defined in the database. timestamp data is treated as binary(8).
- If data in the varchar and varbinary fields is longer than the length specified for copy out, bcp silently truncates the data in the file at the specified length.
- bcp does not know how large any one data value will be before copying all the data, so it always pads char datatypes to their full specified length.

- The file storage type and length of a column do not have to be the same as the type and length of the column in the database table. If the types and formats copied in are incompatible with the structure of the database table, the copy fails.
- File storage length generally indicates the maximum amount of data that can be transferred for the column, excluding terminators and/or prefixes.
- When copying data into a table, bcp observes any defaults defined for columns and user-defined datatypes. However, bcp ignores rules in order to load data at the fastest possible speed.
- bcp considers any data column that can contain a null value to be variable length, so use either a length prefix or a terminator to denote the length of each row of data.
- The file storage type and length of a column need not be the same as the type and length of the column in the database table. (If types and formats copied in are incompatible with the structure of the database table, the copy fails.)

## Field and row terminators

You can use a terminator to mark the end of a column or row, separating one from the next. The default is no terminator.

- Field terminators separate table columns.
- A row terminator is a field terminator for the last field in the row of the table or file.

Terminators are very useful for dealing with character data because you can choose human-readable terminators. The bcp character option, which uses tabs between each column with a newline terminator at the end of each row, is an example of using terminators that enhance the readability of a data file.

When you prepare data for use with other programs, and when you want to use bcp to prepare tabular data, supply your own terminators. The available terminators are:

- Tabs, indicated by `\t`
- New lines, indicated by `\n`
- Carriage returns, indicated by `\r`
- Backslash, indicated by `\`

- Null terminators (no visible terminator), indicated by `\0`
- Any printable character, for example, `*`, `A`, `t`, `|`
- Strings of up to 10 printable characters, including some or all of the terminators listed above (for example, `**\t**`, `end`, `!!!!!!!!!!`, and `\t--\n`)

---

**Note** Control characters (ASCII 0–25) cannot be printed.

---

## Choosing Terminators

Choose terminators with patterns that do not appear in any of the data.

For example, using a tab terminator with a string of data that also contains a tab creates an ambiguity: which tab represents the end of the string? `bcp` always looks for the first possible terminator, which in this case would be incorrect, since the first tab it would encounter would be the one that is part of the data string.

Data in native format can also conflict with terminators. Given a column that contains a 4-byte integer in native format, if the values of these integers are not strictly limited, it will be impossible to choose a terminator that is guaranteed not to appear inside the data. Use `bcp`'s native format option for data in native format.

---

**Note** “No terminator” is different from a “null terminator,” which is an invisible, but real, character.

---

- A field terminator string can be up to 30 characters long. The most common terminators are a tab (entered as `\t` and used for all columns except the last one), and a newline (entered as `\n` and used for the last field in a row). Other terminators are: `\0` (the null terminator), `\` (backslash), and `\r` (Return). When choosing a terminator, be sure that its pattern does not appear in any of your character data, because `bcp` always looks for the first possible terminator.

For example, if you used tab terminators with a string that contained a tab, `bcp` would not be able to identify which tab represents the end of the string. `bcp` always looks for the first possible terminator, so, in this example it would find the wrong one.

A terminator or prefix affects the actual length of data transferred:



When a terminator or prefix is present, it affects the length of data transferred. If the length of an entry being copied out to a file is less than the storage length, it is immediately followed by the terminator or the prefix for the next field. The entry is not padded to the full storage length (char, nchar, and binary data is returned from Adaptive Server already padded to the full length).

When bcp is copying in from a file, data is transferred until either the number of bytes indicated in the “Length” prompt has been copied or the terminator is encountered. Once the number of bytes equal to the specified length has been transferred, the rest of the data is flushed until the terminator is encountered. When no terminator is used, the table storage length is strictly observed.

- Fields stored as char (except char, nchar, and binary fields) instead of their database datatypes take less file storage space with the default length and prefix or a terminator. bcp can use either a terminator or a prefix to determine the most efficient use of storage space. bcp suggests the maximum amount of storage space required for each field as the default. For char or varchar data, bcp accepts any length.
- Table 4-6 and Table 4-7 show the interaction of prefix lengths, terminators, and field length on the information in the file. “P” indicates the prefix in the stored table; “T” indicates the terminator; and dashes, (--) show appended spaces. An ellipsis (...) indicates that the pattern repeats for each field. The field length is 8 bytes for each column; “string” represents the 6-character field each time.

**Table 4-6: Adaptive Server char data**

	<b>Prefix length = 0</b>	<b>Prefix length=1, 2, or 4</b>
No terminator	string--string--...	Pstring--Pstring--...
Terminator	string--Tstring--T...	Pstring--TPstring--T...

**Table 4-7: Other datatypes converted to char storage**

	<b>Prefix length = 0</b>	<b>Prefix length=1, 2, or 4</b>
No terminator	string--string--...	PstringPstring...
Terminator	stringTstringT...	PstringTPstringT...

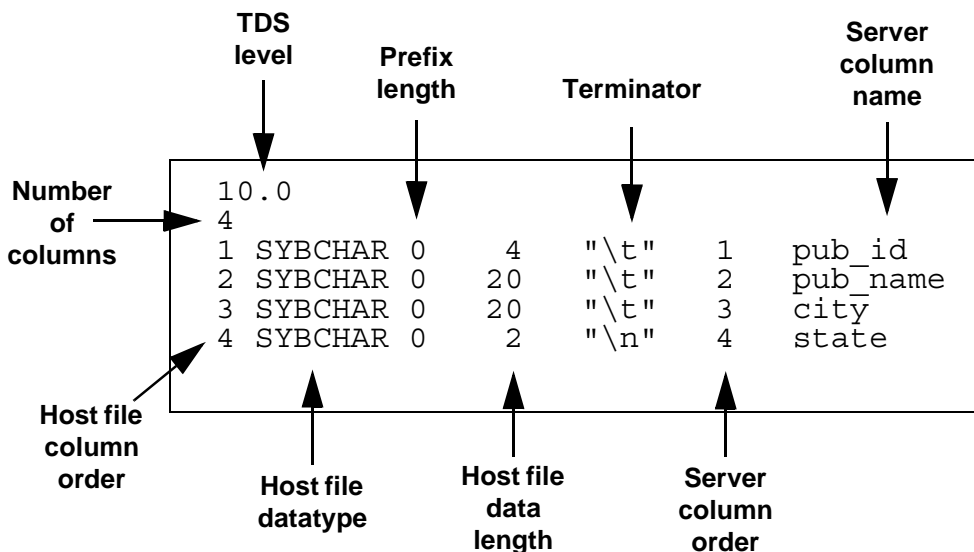
## Using format files

After gathering information about each field in the table, bcp asks if you want to save the information to a **format file** and prompts for the file name.

Using a format file created for the data to be copied with the bcp utility allows you to copy data in or out noninteractively; that is, without being prompted by bcp for information. The format file supplies the information that bcp needs. You can use this newly created format file at any other time to copy the data back into Adaptive Server or to copy data out from the table.

Figure 4-3 illustrates the format of the bcp format files. It shows the publishers table from the pubs2 database, with all the host file columns in character format, with no prefix, and using the default data length, a newline terminator at the end of the final column of a row, and tabs as terminators for all other columns.

Figure 4-3: bcp format file



## Elements of the *bcp* format file

The following list names the various elements of a *bcp* format file. Use Figure 4-3 as the format file example.

- The Tabular Data Stream (TDS) version is always the first line of the file. It specifies the version of TDS that you are using, not the Adaptive Server version, and appears as a literal string without quotation marks. In Figure 4-3, the version is 10.0.
- The second line of a *bcp* format file is the number of columns, which refers to the number of records in the format file, not including lines 1 and 2. Each column in the host table has one line.
- One line for each column follows the first and second lines in the database table. Each line consists of elements that are usually separated by tabs, except for the host file datatype and the prefix length which are usually separated by a space. These elements are:
  - Host file column order
  - Host file datatype
  - Prefix length

- Host file data length
- Terminator
- Server column order
- Server column name
- Column precision
- Column scale

The following sections describe the column elements in the format file.

**Host file column order** The host file column order is the sequential number of the field in the host data file, which begins numbering at 1.

**Host file datatype** The host file datatype refers to the storage format of the field in the host data file, not the datatype of the database table column.

Table 4-8 lists the valid storage formats.

**Table 4-8: Host file datatype storage format**

Storage format	Adaptive Server datatype
SYBCHAR	char / varchar (ASCII)
SYBTEXT	text
SYBBINARY	binary
SYBIMAGE	image
SYBINT1	tinyint
SYBINT2	smallint
SYBINT4	int
SYBFLT8	float
SYBREAL	real
SYBBIT	bit
SYBNUMERIC	numeric
SYBDECIMAL	decimal
SYBMONEY	money
SYBMONEY4	smallmoney
SYBDATETIME	datetime
SYBDATETIME4	smalldatetime

Data written to a host file in its native format preserves all of its precision. datetime and float values preserve all of their precision, even when they are converted to character format. Adaptive Server stores money values to a precision of one ten-thousandth of a monetary unit. However, when money values are converted to character format, their character format values are recorded only to the nearest two places.

See Chapter 1, “System and User-Defined Datatypes” in the *Reference Manual: Building Blocks* for descriptions and appropriate uses of Adaptive Server datatypes.

#### Prefix length

Prefix length indicates the number of bytes in the field length prefix. The prefix length is a 0-, 1-, 2-, or 4-byte unsigned integer value embedded in the host data file that specifies the actual length of data contained in the field. Some fields may have a length prefix while others do not.

Table 4-9 shows the allowable prefix length values.

**Table 4-9: Allowable prefix length values**

Length (in bytes)	Range
0	No prefix
1	$2^8-1$ ; 0-255
2	$2^{16}-1$ ; 0-65535
4	$2^{32}-1$ ; 0-4,294,967,295

Host file data length	<p>Host file data length refers to the maximum number of bytes to copy for the field. To decide how much data to copy in or out, bcp uses one of:</p> <ul style="list-style-type: none"> <li>• The maximum field length</li> <li>• The prefix length, if any</li> <li>• The field terminator string, if any</li> </ul> <p>If more than one method of field length specification is given, bcp chooses the one that copies the least amount of data.</p>
Terminator	The terminator can be up to 30 bytes of characters enclosed in quotation marks (" "). The terminator designates the end of data for the host data file field.
Server column order	The server column order represents the colid (column ID) of the syscolumns column into which the host data file column is to be loaded. Together with the host file column order, this element maps host data file fields to the database table columns.
Server column name	The server column name is the name of the database table column into which this field is to be loaded.
Column precision	The column precision is the precision of the database table column into which this field is to be loaded. This element is present only if the storage format is numeric or decimal.
Column scale	The column scale is the scale of the database table column into which this field is to be loaded. This element is present only if the storage format is numeric or decimal.

## Examples: copying out data interactively

By changing the default values of the prompts to bcp, you can prepare data for use with other software.

To create a human-readable file, respond to the bcp prompts as follows:

- File storage type, enter 0.
- Prefix length, enter 0.
- Field length, accept the default.
- Terminator – the field terminator you enter depends on the software that you plan to use.

- Choose between delimited fields or fixed-length fields. Always use `\n`, the newline terminator, to terminate the last field.

For fixed-length fields, do not use a terminator. Each field has a fixed length, with spaces to pad the fields. Adjacent fields, where the data completely fills the first field seem to run together, since there are no field separators on each line of output. See the example below.

- For comma-delimited output, use a comma (,) as the terminator for each field. To create tabular output, use the tab character (`\t`).

## Copying out data with field lengths

The following example uses fixed-length fields to create output in the personal computer format called SDF (system data format). This format can be easily read or produced by other software.

---

**Note** For information about format files, see “Using format files” on page 72.

---

```
bcp pubs2..sales out sal_out
```

The results as stored in the `sal_out` file are as follows:

```
5023      AB-123-DEF-425-1Z3      Oct 31 1985 12:00AM
5023      AB-872-DEF-732-2Z1      Nov 6 1985 12:00AM
5023      AX-532-FED-452-2Z7      Dec 1 1990 12:00AM
5023      BS-345-DSE-860-1F2      Dec 12 1986 12:00AM
5023      GH-542-NAD-713-9F9      Mar 15 1987 12:00AM
5023      NF-123-ADS-642-9G3      Jul 18 1987 12:00AM
5023      XS-135-DER-432-8J2      Mar 21 1991 12:00AM
5023      ZA-000-ASD-324-4D1      Jul 27 1988 12:00AM
5023      ZD-123-DFG-752-9G8      Mar 21 1991 12:00AM
5023      ZS-645-CAT-415-1B2      Mar 21 1991 12:00AM
5023      ZZ-999-ZZZ-999-0A0      Mar 21 1991 12:00AM
6380      234518                  Sep 30 1987 12:00AM
6380      342157                  Dec 13 1985 12:00AM
6380      356921                  Feb 17 1991 12:00AM
```

```
7066      BA27618      Oct 12 1985 12:00AM
7066      BA52498      Oct 27 1987 12:00AM
7066      BA71224      Aug 5 1988 12:00AM
7067      NB-1.142     Jan 2 1987 12:00AM
7067      NB-3.142     Jun 13 1990 12:00AM
7131      Asoap132     Nov 16 1986 12:00AM
7131      Asoap432     Dec 20 1990 12:00AM
7131      Fsoap867     Sep 8 1987 12:00AM
7896      124152      Aug 14 1986 12:00AM
7896      234518      Feb 14 1991 12:00AM
8042      12-F-9       Jul 13 1986 12:00AM
8042      13-E-7       May 23 1989 12:00AM
8042      13-J-9       Jan 13 1988 12:00AM
8042      55-V-7       Mar 20 1991 12:00AM
8042      91-A-7       Mar 20 1991 12:00AM
8042      91-V-7       Mar 20 1991 12:00AM
```

The contents of the *sal\_fmt* format file are as follows:

```
10.0
3
1      SYBCHAR 04   "" 1  stor_id
2      SYBCHAR 020 "" 2  ord_num
3      SYBCHAR 026 "" 3  date
```

For information about format files, see “Using format files” on page 72.

## Copying out data with delimiters

In the following examples, *bcp* copies data interactively from the publishers table to a file.

---

**Note** For information about format files, see “Using format files” on page 72.

---

## Comma-delimited, newline-delimited with format file

The first example creates an output file with commas between all fields in a row and a newline terminator at the end of each row. This example creates a format file (*pub\_fmt*) that you can use later to copy the same or similar data back into Adaptive Server.

```
bcp pubs2..publishers out pub_out
```



The results as stored in the *pub\_out* file are:

```
0736,New Age Books,Boston,MA
0877,Binnet & Hardley,Washington,DC
1389,Algodata Infosystems,Berkeley,CA
```

The contents of the *pub\_fmt* format file are:

```
10.0
4
1 SYBCHAR 0 4  ", "  1 pub_id
2 SYBCHAR 0 40  ", "  2 pub_name
3 SYBCHAR 0 20  ", "  3 city
4 SYBCHAR 0 2   "\n"  4 state
```

### Tab-delimited with format file

Similarly, the following example creates tab-delimited output from the table *pubs2..publishers* in the *pub\_out* file.

```
bcp pubs2..publishers out pub_out
```

The results as stored in the *pub\_out* file are:

```
0736      New Age Books           Boston      MA
0877      Binnet & Hardley         Washington  DC
1389      Algodata Infosystems       Berkeley    CA
```

The contents of the *pub\_fmt* format file are:

```
10.0
4
1 SYBCHAR 04  "\t"  1 pub_id
2 SYBCHAR 040 "\t"  2 pub_name
3 SYBCHAR 020 "\t"  3 city
4 SYBCHAR 02  "\n"  4 state
```

## Examples: copying in data interactively

To copy in data successfully to a table from a file, you must know what the terminators in the file are or what the field lengths are and specify them when you use *bcp*.

The following examples show how to copy data in, either with fixed field lengths or with delimiters, using bcp with or without a format file.

## Copying in data with field lengths

In this example, bcp copies data from the *salesnew* file into the pubs2..sales table.

In the *salesnew* file are three fields: the first is 4 characters long, the second is 20, and the third is 26 characters long. Each row ends with a newline terminator (\n), as follows:

```
5023ZS-731-AAB-780-2B9      May 24 1993 12:00:00:000AM
5023XC-362-CFB-387-3Z5      May 24 1993 12:00:00:000AM
6380837206                   May 24 1993 12:00:00:000AM
6380838441                   May 24 1993 12:00:00:000AM
```

Use the following command to copy in the data interactively from *salesnew*:

```
bcp pubs2..sales in salesnew
```

The system responds to the bcp command as follows:

```
Password:
Enter the file storage type of field stor_id [char]:
Enter prefix-length of field stor_id [0]:
Enter length of field stor_id [4]:
Enter field terminator [none]:
Enter the file storage type of field ord_num [char]:
Enter prefix-length of field ord_num [1]: 0
Enter length of field ord_num [20]:
Enter field terminator [none]:
Enter the file storage type of field date [datetime]: char
Enter prefix-length of field date [1]: 0
Enter length of field date [26]:
Enter field terminator [none]: \n
Do you want to save this format information in a file? [Y/n] y
Host filename [bcp.fmt]: salesin_fmt
Starting copy...
4 rows copied.
Clock Time (ms.): total = 1 Avg = 0 (116000.00 rows per sec.)
```

When you log in to Adaptive Server and access sales, you see the following data from *salesnew* appended to the table:

```
select * from sales
```

```

stor_id  ord_num                date
-----  -
5023     AB-123-DEF-425-1Z3      Oct 31 1985 12:00AM
5023     AB-872-DEF-732-2Z1      Nov  6 1985 12:00AM

5023     AX-532-FED-452-2Z7      Dec  1 1990 12:00AM
5023     BS-345-DSE-860-1F2      Dec 12 1986 12:00AM
5023     GH-542-NAD-713-9F9      Mar 15 1987 12:00AM
5023     NF-123-ADS-642-9G3      Jul 18 1987 12:00AM
5023     XS-135-DER-432-8J2      Mar 21 1991 12:00AM
5023     ZA-000-ASD-324-4D1      Jul 27 1988 12:00AM
5023     ZD-123-DFG-752-9G8      Mar 21 1991 12:00AM
5023     ZS-645-CAT-415-1B2      Mar 21 1991 12:00AM
5023     ZZ-999-ZZZ-999-0A0      Mar 21 1991 12:00AM
6380     234518                   Sep 30 1987 12:00AM
6380     342157                   Dec 13 1985 12:00AM
6380     356921                   Feb 17 1991 12:00AM
7066     BA27618                  Oct 12 1985 12:00AM
7066     BA52498                  Oct 27 1987 12:00AM
7066     BA71224                  Aug  5 1988 12:00AM
7067     NB-1.142                 Jan  2 1987 12:00AM
7067     NB-3.142                 Jun 13 1990 12:00AM
7131     Asoap132                 Nov 16 1986 12:00AM
7131     Asoap432                 Dec 20 1990 12:00AM
7131     Fsoap867                 Sep  8 1987 12:00AM
7896     124152                   Aug 14 1986 12:00AM
7896     234518                   Feb 14 1991 12:00AM
8042     12-F-9                   Jul 13 1986 12:00AM
8042     13-E-7                   May 23 1989 12:00AM
8042     13-J-9                   Jan 13 1988 12:00AM
8042     55-V-7                   Mar 20 1991 12:00AM
8042     91-A-7                   Mar 20 1991 12:00AM
8042     91-V-7                   Mar 20 1991 12:00AM
(34 rows affected)

```

Since there is a unique clustered index on the stor\_id and ord\_num columns of sales, the new rows were sorted in order.

A conflict or violation can affect the copy process:

- Had there been any violations of the unique index on the columns in the data being copied from the file, bcp would have discarded the entire batch in which the violating row was encountered.

A batch size of 1 evaluates each row individually, but loads more slowly and creates a separate data page for each row during a fast bcp session.

- If the types copied in are incompatible with the database types, the entire copy fails.

## Copying in data with delimiters

In the following example, `bcp` copies data from the file `newpubs` into the table `pubs2.publishers`. In the `newpubs` file, each field in a row ends with a tab character (`\t`) and each row ends with a newline terminator (`\n`), as follows:

```
1111 Stone Age Books           Boston      MA
2222 Harley & Davidson        Washington  DC
3333 Infodata Algosystems      Berkeley    CA
```

Since `newpubs` contains all character data, you can use the character command-line flag and specify the terminators with command line options:

- In UNIX platforms:

```
bcp pubs2..publishers in newpubs -c -t\t -r\n
```

- In Windows:

```
bcp pubs2..publishers in newpubs -c -t\t -r\n
```

## Copying in data with a format file

To copy data back into Adaptive Server using the saved `pub_fmt` format file, run the following command:

```
bcp pubs2..publishers in pub_out -fpub_fmt
```

You can use the `pub_fmt` file to copy any data with the same format into Adaptive Server. If you have a similar data file with different delimiters, you can change the delimiters in the format file.

Similarly, you can edit the format file to reflect any changes to the field lengths, as long as all fields have the same length. For example, the `moresales` file contains:

```
804213-L-9 Jan 21 1993 12:00AM
804255-N-8 Mar 12 1993 12:00AM
804291-T-4 Mar 23 1993 12:00AM
804291-W-9 Mar 23 1993 12:00AM
```

Edit the `sal_fmt` format file to read as follows:

```
10.0
3
1 SYBCHAR 0 4 "" 1 stor_id
2 SYBCHAR 0 7 "" 2 ord_num
3 SYBCHAR 0 21 "\n" 3 date
```

Then enter the following command:

- In UNIX platforms:

```
bcp pubs2..sales in moresales -fsal_fmt
```

- In Windows:

```
bcp pubs2..sales in moresale -fsal_fmt
```

The system responds as follows:

```
Starting copy...
4 rows copied.
Clock Time (ms.): total = 1 Avg = 0 (116000.00 rows
per sec.)
```

## Using *bcp* with alternate languages

Adaptive Server stores data using its default character set, which is configured during installation. If your terminal does not support that default character set, it may send confusing characters to *bcp* when you respond to prompts either by typing or by using host file scripts.

Omitting all character-set options causes *bcp* to use the character set that was named as the default for the platform. This default can cause communications problems:

- The default is not necessarily the same character set that was configured for Adaptive Server.
- The default may not necessarily be the character set that the client is using.

For more information about character sets and the associated flags, see Chapter 8, “Configuring Client/Server Character Set Conversions,” in the *System Administration Guide*.

## **bcp and row-level access rules**

If Adaptive Server is enabled for row-level access, and you bulk-copy-out data, bcp copies out only the rows of data to which you have access. To copy out the entire table, you must first drop the access rules, then bcp out. Reinststate the access rules after you are done, if applicable.

If you bulk-copy-in data to a table that has access rules enabled, Adaptive Server may issue “uniqueness violation” errors. For example, if you load data from a bcp data file that was generated before the access rules were created on the table, and the bcp data file contains rows that were previously inserted into the table, you may receive this type of error.

If this happens, the table may look to the user like it does not include the rows that failed the bcp insert because of the uniqueness violation, but the user does not have access to the “missing” rows because of the access rules.

To copy in the entire table, drop the access rules, load the data, address any errors, then reinststate the access rules.

## **Copy in and batch files**

Batching applies only to bulk copying in; it has no effect when copying out. By default, Adaptive Server copies all the rows in batches of 1000 lines. To specify a different batch size, use the command-line option (-b).

bcp copies each batch in a single transaction. If Adaptive Server rejects any row in the batch, the entire transaction is rolled back. By default, bcp copies all rows in a single batch; use the -b parameter to change the default batch size. Adaptive Server considers each batch a single bcp operation, writes each batch to a separate data page, and continues to the next batch, regardless of whether the previous transaction succeeded.

When data is being copied in, it can be rejected by either Adaptive Server or bcp.

- Adaptive Server treats each batch as a separate transaction. If the server rejects any row in the batch, it rolls back the entire transaction.
- When bcp rejects a batch, it then continues to the next batch. Only fatal errors roll back the transaction.

- Adaptive Server generates error messages on a batch-by-batch basis, instead of row-by-row, and rejects each batch in which it finds an error. Error messages appear on your terminal and in the error file.

## Improving recoverability

To ensure better recoverability:

- Break large input files into smaller units.

For example, if you use bcp with a batch size of 100,000 rows to bulk copy in 300,000 rows, and a fatal error occurs after row 200,000, bcp would have successfully copied in the first two batches—200,000 rows—to Adaptive Server. If you had not used batching, bcp would not have been able to copy in any rows to Adaptive Server.

- Set the trunc log on chkpt to true (on).

The log entry for the transaction is available for truncation after the batch completes. If you copy into a database that has the trunc log on chkpt database option set on (true), the next automatic checkpoint removes the log entries for completed batches. This log cleaning breaks up large bcp operations and keeps the log from filling.

- Set `-b batch_size` to 10.

The batch size parameter set to 10 causes bcp to reject the batch of 10 rows, including the defective row. The error log from this setting allows you to identify exactly which row failed.

A batch size of 1 is the smallest that bcp processes.

---

**Note** bcp creates 1 data page per batch, and setting `b batch_size` to 10 creates data pages with 10 rows on each page. If you set `-b batch_size` to 1, the setting creates data pages with 1 row on each page. This setting causes the data to load slowly and takes up storage space.

---

## Batches and partitioned tables

When you bulk copy data into a partitioned table without specifying a partition number, Adaptive Server randomly assigns each batch to an available partition. Copying rows in a single batch places all those rows in a single partition, which can lead to load imbalance in the partitioned table.

To help keep partitioned tables balanced, use a small batch size when bulk copying data or specify the partition ID during the bcp session. For information about partitioning tables, see the *Performance and Tuning Guide*.

## Copy out and text and image data

When you copy out text or image data, Adaptive Server, by default, copies only the first 32K of data in a text or image field. The `-T text_or_image_size` parameter allows you to specify a different value. For example, if the text field to copy out contains up to 40K of data, you can use the following command to copy out all 40K:

```
bcp pubs2..publishers out -T40960
```

---

**Note** If a text or image field is larger than the given value or the default, bcp does not copy out the remaining data.

---



## Specifying a network packet size

To improve the performance of large bulk copy operations, you may want to use larger network packet sizes than the defaults. The `-A size` option specifies the network packet size to use for the bcp session that you are beginning.

The value of *size* must be:

- Between the values of the default network packet size and max network packet size configuration parameters, and
- A multiple of 512.

---

**Note** The new packet size remains in effect for the current bcp session only.

---

For example, this command specifies that Adaptive Server send 40K of text or image data using a packet size of 4096 bytes for the bcp session:

```
bcp pubs2..authors out -A 4096 -T40960
```

## Copy in and error files

When you specify the `-e error_file` option with copy in, bcp stores the rows that it cannot copy in to Adaptive Server in the specified error file.

- The error file stores:
  - A line that indicates which row failed and the error that occurred, and
  - A line that is an exact copy of the row in the host file.
- If the file name specified after `-e` already exists, bcp overwrites the existing file.
- If bcp does not encounter any errors, it does not create the file.

bcp in detects two types of errors:

- Data conversion errors
- Errors in building the row; for example, attempts to insert a NULL into columns that do not accept null values or to use invalid data formats, such as a 3-byte integer

The copy in process displays error messages on your monitor.

The following example loads the *newpubs* file into the publishers database, storing any error rows in the *pub\_err* file:

```
bcp pubs2..publishers in newpubs -epub_err
```

Keep the following in mind when working with error files generated by copy in:

- bcp stores rows in an error file only when the bcp program itself detects the error.
- bcp continues to copy rows until bcp encounters the maximum number of error rows, at which point bcp stops the copy.
- bcp sends rows to Adaptive Server in batches, so bcp cannot save copies of rows that are rejected by Adaptive Server, for example, a duplicate row for a table that has a unique index.
- Adaptive Server generates error messages on a batch-by-batch basis, instead of row-by-row, and rejects the entire batch if it finds an error.
- It is not considered an error for Adaptive Server to reject duplicate rows if either `allow_dup_row` or `ignore_dup_key` was set when a table's index was created. The copy proceeds normally, but the duplicate rows are neither stored in the table nor in the bcp error file.

## Copy out and error files

During the copy out process, as with copy in, bcp overwrites any file of the same name and does not create an error file if no errors occurred.

There are two situations that cause rows to be logged in the error file during a copy out:

- A data conversion error in one of the row's columns
- An I/O error in writing to the host file

Keep the following in mind when working with error files generated by copy out:

- bcp logs rows in the error file in the default character format.
- All data values print as characters with tabs between the columns and a newline terminator at the end of each row.

## Data integrity: defaults, rules, and triggers

To ensure integrity, *bcp* handles data to copy depending upon its element.

### Defaults and datatypes

When copying data into a table, *bcp* observes any defaults defined for the columns and datatypes. That is, if there is a null field in the data in a file, *bcp* loads the default value instead of the null value during the copy.

For example, here are two rows in a file to be loaded into authors:

```
409-56-7008,Bennet,David,415 658-9932,622 Pine
St.,Berkeley,CA,USA,94705213-46-8915,Green,Marjorie,,309 63rd St.
#411,Oakland,CA,USA,94618
```

Commas separate the fields; a newline terminator separates the rows. There is no phone number for Marjorie Green. Because the phone column of the authors table has a default of “unknown,” the rows in the loaded table look like this:

```
409-56-7008  Bennet  David      415 658-9932  622 Pine St.
              Berkeley CA  USA  94705
213-46-8915  Green   Marjorie  unknown      309 63rd St. #411
              Oakland CA  USA  94618
```

### Rules and triggers

*bcp*, to enable its maximum speed for loading data, does not fire rules and triggers.

To find any rows that violate rules and triggers, copy the data into the table and run queries or stored procedures that test the rule or trigger conditions.

## How *bcp* differs from other utilities

The *bcp* utility, which copies entire tables or portions of a single table, is distinct from the other utilities that move data from one place to another.

The following list names these other utilities and their commands and describes how you can best use them to move data.

*dump database, load database, dump transaction, and load transaction*

Use the SQL commands dump database, load database, dump transaction, and load transaction for backup purposes only. Unlike bcp, the dump commands create a physical image of the entire database.

You must use load database or load transaction to read data backed up with dump database or dump transaction.

For information on using the SQL dump and load commands, see the *System Administration Guide* and the *Reference Manual*.

*insert, update, and delete*

Use the data modification commands insert, update, and delete, respectively, to add new rows to, change existing rows in, or remove rows from a table or view.

- Use the insert command with a select statement to move data between tables.
- Use the select statement with an into clause to create a new table, based on:
  - the columns named in the select statement,
  - the tables named in the from clause, and
  - data in the rows named in the where clause.

For details on adding, changing, and deleting data, see insert, update, and delete in the *Reference Manual*.

# Using dsedit

This chapter explains how to use the dsedit utility to edit the Adaptive Server interfaces file.

Topic	Page
Getting started with dsedit	91
Adding, viewing, and editing server entries	95
Troubleshooting dsedit	102

## Getting started with dsedit

dsedit is a graphical utility that lets you view and edit server entries in the interfaces file (*sql.ini* in Windows). For a detailed description of dsedit syntax, see dsedit on page 193.

---

**Note** UNIX users: If your system does not have X-Windows, use dscpc to configure server entries in the interfaces file. See Chapter 6, “Using dscpc” for more information.

---

## Starting dsedit

Windows

You can start dsedit from the command prompt, the Windows Explorer, or the Sybase for Windows program group.

### v Starting dsedit from the command prompt

- Enter:

```
dsedit
```

You can specify the following command-line arguments:

- `-ddsname` – Specifies which directory service to connect to. *dsname* is the local name of the directory service, as listed in the *libtcl.cfg* file. If you do not specify the `-ddsname` argument, dsedit presents a list of directory service options in the first dialog box.
- `-lpath` – Specifies the path to the *libtcl.cfg* file, if other than *SYBASE\_home\INI*. Use this argument only if you want to use a *libtcl.cfg* file other than the one located in *SYBASE\_home\INI*.

v **Starting dsedit through the Windows Explorer**

- 1 Go to the `%SYBASE%\bin\` directory.
- 2 Double-click on the *DSEEDIT.exe* file.

v **Starting dsedit from the Sybase for Windows program group**

- 1 Choose Sybase for Windows from the Start menu.
- 2 Choose dsedit from the Sybase for Windows menu. The Select Directory Service dialog box appears.

UNIX platforms

Before starting dsedit, make sure that you have write permission on the *interfaces* file.

If you are running dsedit from a remote machine, make sure that the `DISPLAY` environment variable is set so the dsedit screens will show on your machine instead of on the remote machine.

v **Setting the DISPLAY environment variable**

- 1 Log in to the remote machine.
- 2 Enter:

```
setenv DISPLAY your_machine_name:0.0
```

v **Starting dsedit**

- Enter:

```
SYBASE/bin/dsedit
```

The Select a Directory Service window appears. This window lets you open editing sessions for the *interfaces* file. The full path name of the default *interfaces* file is shown in the Interfaces File to Edit box. The full path name of the configuration file is shown below it.

## Opening an editing session

### Windows

The Select Directory Service dialog box allows you to open a session with a directory service. You can open a session with:

- Any directory service that has a driver listed in the *libtcl.cfg* file
- The *sql.ini* file

#### v **Opening a session in Windows**

- 1 Double-click on the local name of the directory service you want to connect to, as listed in the DS Name box, or
- 2 Click on the local name of the directory service you want to connect to, as listed in the DS Name box, and click the OK button.

---

**Note** dsedit uses the SYBASE environment variable to locate the *libtcl.cfg* file. If the SYBASE environment variable is not set correctly, dsedit cannot locate the *libtcl.cfg* file.

---

The session number and local name of the directory service appear in the header bar.

#### v **Opening additional sessions**

dsedit allows you to have multiple sessions open at one time.

- 1 Choose Open Directory Service from the File menu.

The Select Directory Service box appears.

- 2 Double-click the local name of the directory service to which you want to be connected (or click on the directory service name and click OK).

Opening multiple sessions allows you to copy entries between directory services. See “Copying server entries” on page 100 for more information.

#### v **Switching between sessions**

If you have multiple sessions open at one time, you need to activate a session before you can work in it.

- Activate a session by either:
  - Clicking in the session window
  - Choosing the session from the Windows menu

The dsedit title bar shows which session is active.

## UNIX platforms

### v **Opening the default interfaces file for editing**

- 1 Select Sybase Interfaces File.
- 2 Click OK.

### v **Opening a file other than the default interfaces file**

- 1 Select Sybase Interfaces File.
- 2 Edit the displayed file name.
- 3 Click OK.

The Directory Service Session window appears.

You can open multiple interfaces file sessions with different files.

The Directory Service Session screen displays the full path name of the interfaces file and lists the server entries contained within it.

- **Add new server entry** – displays the Server Entry Editor window, where you specify the name and network addresses for a new server entry.
- **Modify server entry** – lets you view and modify the network addresses for a selected server entry. To view or modify a server entry, select the server in the list, then click Modify server entry to display the server's attributes in the Server Entry Editor window.
- **Copy server entry** – lets you copy one or more entries to another interfaces file.
- **Close Session** – closes the session window and writes changes to the interfaces file.

For procedures on using these buttons, see “Modifying server entries in Windows” on page 95.

Clicking the Add new server entry or Modify server entry button in the Session screen displays the Server Entry Editor window.

You use the Server Entry Editor window to view or edit server entries in an interfaces file:

- **Server name** – if you are adding a server entry, type the name of the new server. If you are editing a server entry, you can edit the name field to rename the server. The new name cannot already exist in the interfaces file.



- Available network transports – a list of the network addresses where the server accepts client connections.
  - To create a new address, click Add network transport. See “Modifying server entries in Windows” on page 95.
  - To edit an existing address, click Modify network transport. See “Modifying server entries in Windows” on page 95.
  - To remove a selected network address, click Delete network transport.
  - To rearrange the order of addresses in the list, click Move network transport up or Move network transport down.
- OK – commits your changes and closes the window. Changes to the interfaces file are not applied until you close the session using the Close Session button in the Directory Service Session screen.
- Cancel – closes the window and discards any edits.

## Adding, viewing, and editing server entries

Once you are in an open session, you can add, modify, rename and delete server entries associated with that session, as well as copy server entries within a session and between sessions.

### Modifying server entries in Windows

The server entries associated with the session appear in the Server box. Click on a server entry to select it.

Each server entry is made up of a set of attributes. The attributes are described in Table 5-1.

**Table 5-1: Server attributes**

Attribute name	Type of value	Description	Default value
Server Version	Integer	Version level of the server object definition. Sybase provides this attribute to identify future changes to the object definition.	150
Server Name	Character string	Server name.	N/A
Server Service	Character string	A description of the service provided by the server. This value can be any meaningful description.	Adaptive Server

Attribute name	Type of value	Description	Default value
Server Status	Integer	The operating status of the server. Values are: <ul style="list-style-type: none"> <li>• Active</li> <li>• Stopped</li> <li>• Failed</li> <li>• Unknown</li> </ul>	4
Security Mechanism	Character string	Object identifier strings (OID) that specify the security mechanisms supported by the server. This attribute is optional. If it is omitted, Open Server allows clients to connect with any security mechanism for which Open Server has a corresponding security driver.	N/A
Server Address	Character string	One or more addresses for the server. The format of the address varies by protocol, and some protocols allow more than one format. The options are: <ul style="list-style-type: none"> <li>• TCP/IP – two formats: <ul style="list-style-type: none"> <li>• <i>computer name,port number</i></li> <li>• <i>ip-address.portnumber</i></li> </ul> </li> <li>• Named Pipe – <i>pipe name</i>: “\pipe” is a required prefix to all pipe names. Server pipes can be only local. <ul style="list-style-type: none"> <li>• <b>Local</b> – <i>\pipe\sql\query</i></li> <li>• <b>Remote</b> – <i>\\computer_name\pipe\sql\query</i></li> </ul> </li> <li>• IPX/SPX – three formats: <ul style="list-style-type: none"> <li>• <i>server name</i></li> <li>• <i>net number,node number,socket number</i></li> <li>• <i>server name,socket number</i></li> </ul> </li> <li>• DECnet – four formats: <ul style="list-style-type: none"> <li>• <i>area number.node number,object name</i></li> <li>• <i>area number.node number,object number</i></li> <li>• <i>node name,object name</i></li> <li>• <i>node name,object number</i></li> </ul> </li> </ul>	N/A

v **Adding a server entry**

- 1 Choose Server Object | Add.
- 2 Type a server name in the Server Name box.
- 3 Click OK.

The server entry appears in the Server box. To specify an address for the server, you must modify the entry.

**v Modifying a server attribute**

You can modify any attribute of a server entry.

- 1 Click on a server entry in the Server box.
- 2 Choose Server Object | Modify Attribute.
- 3 Click on the attribute you want to modify in the Attributes box.  
A dialog box appears that shows the current value of the attribute.
- 4 Type a new value for the attribute, or select a value from the drop-down list.

See Table 5-1 on page 95 for a description of each attribute.

- 5 Click OK.

**v Renaming a server entry**

- 1 Click on a server entry in the Server box.
- 2 Choose Server Object | Rename.
- 3 Type a new name for the server entry in the Server Name box.
- 4 Click OK.

**v Deleting a server entry**

- 1 Click on a server entry in the Server box.
- 2 Choose Server Object | Delete.

**v Copying server entries within the current session**

- 1 Click on one or more server entries in the Server box.  
Use the Shift key to select multiple entries.
- 2 Click the Copy button (below the menu bar), or choose Edit | Copy.
- 3 Click the Paste button (below the menu bar), or choose Edit | Paste.

dsedit appends the copied server entries with a version number of *\_n*. You can rename the copied server entries Server Object | Rename option on. See “Renaming a server entry” on page 97 for more information.

**v Copying server entries between sessions**

- 1 Open a session with the directory service or *sql.ini* file that you want the entries copied to.

- 2 To open a session, choose File | Open Directory Service. See “Opening additional sessions” on page 93 for more information.
  - 3 Click on one or more server entries in the Server box of the session that you want the entries copied from.  
Use the Shift key to select multiple entries.
  - 4 To copy the server entries, click the Copy button (below the menu bar), or choose Edit | Copy.  
To cut the server entries, click the Cut button (below the menu bar), or choose Edit | Cut.
  - 5 Activate the session where you want to paste the server entries.  
See “Switching between sessions” on page 93 for instructions for activating a session.
  - 6 Click the Paste button (below the menu bar), or choose Edit | Paste.
- You can rename the copied server entries using Server Object | Rename. See “Switching between sessions” on page 93 for more information.

## Modifying server entries in UNIX platforms

To perform the procedures in this section, open the interfaces session window using the instructions in “Opening an editing session” on page 93.

---

**Note** After performing each procedure in this section, you must click on Close Session to apply your edits to the interfaces file. Clicking this button also closes the interfaces session window.

---

### v **Adding a new server entry**

- 1 Click on Add new server entry.
- 2 Specify the name and network addresses for a new server entry.

### v **Viewing or modifying a server entry**

- 1 Click on Modify server entry.
- 2 Modify the attributes as desired.

### v **Copying a server entry to another interfaces file**

- 1 Use one of the following methods to select the entries to copy:

- To copy a single entry, click it once.
  - To copy a range of consecutive entries, click the first entry in the range, press and hold down Shift, and click the last entry in the range. You can also select “backwards” by clicking the last entry, holding down Shift, and clicking the first entry.
  - To select multiple, nonconsecutive entries, press and hold down the Ctrl key while you click each entry.
- 2 Click Copy server entry.
  - 3 Select the Sybase interfaces file from the list.
  - 4 Edit the displayed file name.
  - 5 Click OK.

## Adding or editing network transport addresses

The Network Transport Editor window allows you to view, edit, or create the transport addresses at which a server accepts client connections. This window displays the name of the server entry for the address and allows you to configure the following items:

- Transport type – specifies the protocol and interface for the address. For all platforms except Digital UNIX, values are tcp, tli tcp, tli spx, and spx. For Digital UNIX, values are decnet, tcp, and tli tcp.
- Address information – depending on the transport type, different address components are required. The following sections discuss address formats in detail.

### TCP/IP addresses

The address information for a TCP/IP entry consists of a host name (or IP address) and a port number (entered as a decimal number). For tli tcp-formatted interfaces entries, the host’s IP address and the port number are converted to the 16-byte hexadecimal representation required for tli tcp-formatted interfaces entries.

In interfaces entries, use tli tcp for:

- All pre-10.0 clients on platforms that use tli-formatted interfaces entries
- Adaptive Server or Replication Server version 11.0.x or earlier on platforms that use tli-formatted interfaces entries

Use tcp for other clients and servers.

To indicate a TCP/IP address, choose `tcp` or `tli tcp` from the Transport Type menu.

## SPX/IPX addresses

SPX/IPX addresses allow Adaptive Server to listen for connections from client applications running on a Novell network. SPX/IPX addresses consist of the following information:

- Host address – an eight digit hexadecimal value representing the IP address of the computer on which the server runs. Each component of the dot-separated decimal IP address format maps to one byte in the hex address format. For example, if your host’s IP address is 128.15.15.14, enter “800F0F0E” as the SPX/IPX host address value.
- Port number – the port number, expressed as a four-digit hexadecimal number.
- Endpoint – the path for the device file that points to the SPX device driver. Defaults to `/dev/mspx` on Solaris and `/dev/nspx` on any other platform. If necessary, adjust the path so that it is correct for the machine on which the server runs. The default path is based on the platform on which you are running `dsedit`.

To indicate an SPX/IPX address, choose `tli spx` or `spx` from the Transport Type menu.

## Copying server entries

`dsedit` allows you to copy server entries within a session and between sessions. This includes copying entries from a `sql.ini` file to a directory service.

Windows

### v Copying server entries within the current session

- 1 Click on one or more server entries in the Server box.  
Use the Shift key to select multiple entries.
- 2 Click the Copy button (below the menu bar), or choose Edit | Copy.
- 3 Click the Paste button (below the menu bar), or choose Edit | Paste.

`dsedit` appends the copied server entries with a version number of `_n`. You can rename the copied server entries using Server Object | Rename. See “Renaming a server entry” on page 97 for more information.

v **Copying server entries between sessions**

- 1 Open a session with the directory service or *sql.ini* file that you want the entries copied to.
- 2 To open a session, choose File | Open Directory Service. See “Opening additional sessions” on page 93 for more information.
- 3 Click on one or more server entries in the Server box of the session that you want the entries copied from.

Use the Shift key to select multiple entries.

- 4 To copy the server entries, click the Copy button (below the menu bar), or choose Edit | Copy.

To cut the server entries, click the Cut button (below the menu bar), or choose Edit | Cut.

- 5 Activate the session where you want to paste the server entries.

See “Switching to another open session” on page 106 for instructions for activating a session.

- 6 Click the Paste button (below the menu bar), or choose Edit | Paste.

You can rename the copied server entries using the Rename command in the Server Object menu. See “Renaming a server entry” on page 97 for more information.

## UNIX platforms

v **Copying a server entry to another interfaces file**

- 1 Use one of the following methods to select the entries to copy:
  - To copy a single entry – click it once.
  - To copy a range of consecutive entries – click the first entry in the range, press and hold down Shift, and click the last entry in the range. You can also select “backwards” by clicking the last entry, holding down Shift, and clicking the first entry.
  - To select multiple, nonconsecutive entries – press and hold down the Ctrl key while you click each entry.
- 2 Click Copy server entry.
- 3 Select the Sybase interfaces file from the list.
- 4 Edit the displayed file name.
- 5 Click OK.

## Troubleshooting dsedit

This section lists some common dsedit problems and describes how to correct them.

### The dsedit utility does not start

Check for the following:

- The SYBASE environment variable is not set or points to the wrong directory.
- **UNIX platforms** X-Windows is not configured correctly. If you are running dsedit on a remote host, make sure that X-Windows clients on the remote host can connect to the X-Windows server on your own machine. See your X-Windows documentation for more troubleshooting information. If X-Windows is not available, use dscp instead of dsedit.

### Error message: “Unable to open X display”

**UNIX platforms** dsedit might not work if the display machine is set up to reject X-Windows connections from remote hosts. If this is the problem, you see a message similar to the following:

```
Unable to open X display. Check the value of your
$DISPLAY variable. If it is set correctly, use the
'xhost +' command on the display machine to authorize
use of the X display. If no X display is available, run
dscp instead of dsedit.
```

This error may be caused by either of the following situations:

- The value for the DISPLAY environment variable is not entered correctly or is not set.

**Solution:** Enter the DISPLAY environment variable correctly.

- You are not authorized to open windows on the machine to which DISPLAY refers.

**Solution:** Run the command 'xhost +' on the display machine.



## **Cannot add, modify, or delete server entries**

Check for permissions problems with the interfaces file. To edit interfaces entries, you must have write permission on both the interfaces file and the Sybase installation directory.



# Using dscp

dscp is a utility program that you use to view and edit server entries in the interfaces file.

---

**Note** dscp is not available for Windows.

---

Topic	Page
Getting started with dscp	105
Working with server entries	107
Exiting dscp	112
Quick reference for dscp utility commands	112

For a detailed description of dscp syntax, see dscp on page 192.

## Getting started with dscp

### v Starting dscp

- Enter:

```
$SYBASE/$SYBASE_OCS/bin/dscp
```

The dscp prompt, >>, appears.

### v Getting help with dscp

- To view the dscp help screen, enter one of the following commands:

```
help  
h  
?
```

## Using a dscp session

Before you can view, add, or modify server entries, you must open a session so that you can interact with the interfaces file.

You can have multiple sessions open at one time.

### v **Opening a session with the interfaces file**

- Enter:

```
open InterfacesDriver
```

When you open a session, dscp provides the session's number. For example, if you open a session using the open InterfacesDriver command, dscp displays the following message:

```
ok  
Session 1 InterfacesDriver>>
```

### v **Listing all open sessions**

- Enter:

```
sess
```

### v **Switching to another open session**

- Enter the following, where *sess* is the session number:

```
switch sess
```

For example, you are switched to session 3 if you enter:

```
switch 3
```

The switch keyword is optional. For example, entering “3” also switches you to session 3.

### v **Closing a session**

- Enter the following, where *sess* is the session number:

```
close sess
```

For example, session 3 closes if you enter:

```
close 3
```

If you do not specify a session number, dscp closes the current session.

## Working with server entries

Use dscp to add or modify server entries.

### Adding and modifying server entries

After you open a session, you can add or modify server entries associated with that session.

---

**Note** When you add or modify a server entry, dscp automatically creates or modifies both master and query lines. The master line and the query line of an interfaces file entry contain identical information.

---

Each server entry is made up of a set of attributes. When you add or modify a server entry, dscp prompts you for information about each attribute. Table 6-2 describes each attribute.

**Table 6-1: Server attributes**

Attributes	Type of value	Default value and valid values	Can be edited when adding or modifying a server entry	
			Adding	Modifying
Server Object Version	Integer	110	No	No
Server Name	Character string	N/A	N/A	No
Server Service	Character string	SQL SERVER	Yes	No
Server Status	Integer	4 Valid values are: 1 Active 2 Stopped 3 Failed 4 Unknown	No	No
Transport Type	Character string	tcp. Valid values are: decnet, spx, tcp, tli, spx, tli tcp	Yes	Yes
Transport Address	Character string	None. Valid values are character strings recognized by the specified transport type	Yes	Yes

Attributes	Type of value	Default value and valid values	Can be edited when adding or modifying a server entry	
			Adding	Modifying
Security Mechanism	Character string	None	Yes	Yes
	<b>Note</b> You can add up to 20 security mechanism strings for each server entry	Valid values are character strings associated with object identifiers defined in the user's <i>objectid.dat</i> .		

v **Adding a server entry**

1 Enter:

```
add servername
```

You are now in add mode. You can continue to add server entries, but you cannot execute any other dscp commands until you exit this mode. While in add mode, dscp prompts you for information about *servername*.

2 Do one of the following:

- Enter a value for each attribute, or
- Press Return to accept the default value, which is shown in brackets [ ]

For example, dscp prompts for the following information when you enter:

```
add myserver
Service: [SQL Server]
Transport Type: [tcp] tcp
Transport Address: victory 8001
Security Mechanism []:
```

A server entry can have up to 20 transport type/address combinations associated with it.

For a description of the server attributes, see Table 6-1 on page 107.

3 To exit add mode, enter:

```
#done
```

v **Modifying a server entry**

You cannot use dscp to modify the Version, Service, and Status entries in the *interfaces* file.

1 Enter:

```
mod servername
```

You are now in modify mode. You can continue to modify server entries, but you cannot execute any other dscp commands until you exit this mode. In modify mode, dscp prompts you for information about *servername*.

- 2 Do one of the following:
  - Enter a value for each attribute, or
  - Press Return to accept the default value, which is shown in brackets [ ]

For example, dscp prompts for the following information when you enter:

```
mod myserver
Version: [1]
Service: [SQL Server] Open Server
Status: [4]
Address:
Transport Type: [tcp]
Transport Address: [victory 1824] victory 1826
Transport Type: [tcp]
Transport Address: [victory 1828]
Transport Type: []
Security Mechanism []:
```

For a description of the server attributes, see Table 6-1 on page 107.

- 3 To delete an address, enter:
 

```
#del
```
- 4 To exit modify mode, enter:
 

```
#done
```

## Copying server entries

dscp allows you to copy server entries within a session and between two sessions. You have four options when copying a server entry.

You can copy:

- A server entry to a new name in the current session
- A server entry to a different session
- A server entry to a new name in a different session

- All entries in the current session to a different session

v **Creating a new server entry within a session by copying**

- Enter:

```
copy name1 to name2
```

For example, if you enter the following, dscp creates a new entry, “my\_server,” that is identical to “myserver”:

```
copy myserver to my_server
```

You can then modify the new entry and leave the original intact.

v **Copying a server entry without changing the name**

- Enter:

```
copy name1 to sess
```

For example, dscp copies the “myserver” entry in the current session to session 2 when you enter:

```
copy myserver to 2
```

v **Copying a server entry and rename it**

- Enter:

```
copy name1 to sess name2
```

For example, dscp copies the “myserver” entry in the current session to session 2 and renames it “my\_server” when you enter:

```
copy myserver to 2 my_server
```

v **Copying all entries in the current session to a different session**

- Enter:

```
copyall sess
```

For example, dscp copies all entries in the current session to session 2 when you enter:

```
copyall 2
```

## Listing and viewing contents of server entries

You can list names and attributes associated with a session.



v **Listing names of server entries**

- Enter:

```
list
```

v **Listing the attributes of server entries**

- Enter:

```
list all
```

For a description of server attributes, see Table 6-1 on page 107.

v **Viewing the contents of a server entry**

- Enter:

```
read servername
```

For example, the following information is displayed when you enter:

```
read myserver

DIT base for object: interfaces
Distinguish name: myserver
Server Version: 1
Server Name: myserver
Server Service: SQL Server
Server Status: 4 (Unknown)
Server Address:
Transport Type: tcp
Transport Addr: victory 1824
Transport Type: tcp
Transport Addr: victory 1828
```

For a description of the server attributes, see Table 6-1 on page 107.

## Deleting server entries

You can delete one entry or all entries associated with a session.

v **Deleting entries associated with a session**

- Enter:

```
del servername
```

For example, dscp deletes the entry for “myserver” when you enter:

```
del myserver
```

v **Deleting all entries associated with a session**

- Enter:  

```
delete-all
```

## Exiting dscp

To exit dscp, enter one of the following commands:

```
exit
quit
```

## Quick reference for dscp utility commands

dscp allows you to perform functions by entering commands at the dscp prompt. Table 6-2 provides a quick reference to these commands.

**Table 6-2: dscp commands**

Command	Description
add <i>servername</i>	Adds server entry <i>servername</i> in the current session. dscp prompts you for information about <i>servername</i> . Press Return to accept the default value, which is shown in square brackets [ ]. Enter “#done” to exit add mode.
addattr <i>servername</i>	Adds an attribute to the server entry <i>servername</i> in the current session.
close [ <i>sess</i> ]	Closes a session identified by the <i>sess</i> number. If you do not specify <i>sess</i> , closes the current session.
config	Displays configuration information related to your Sybase environment.
copy <i>name1</i> to { <i>name2</i>   <i>sess</i>   <i>sess name2</i> }	Copies server entry <i>name1</i> in the current session to: <ul style="list-style-type: none"> <li>• Server entry <i>name2</i> in the current session,</li> <li>• Session <i>sess</i>, or</li> <li>• Server entry <i>name2</i> in session <i>sess</i>.</li> </ul>
copyall to <i>sess</i>	Copies all server entries in the current session to session <i>sess</i> .
del <i>servername</i>	Deletes server entry <i>servername</i> in the current session.
delete-all	Deletes all server entries in the current session.
exit	Exits dscp.
help, ?, h	Displays the online help.

---

<b>Command</b>	<b>Description</b>
list [all]	Lists the server entries for the current session. To list the names of the entries, use the list command. To list the attributes for each entry, use the list all command.
mod <i>servername</i>	Modifies server entry <i>servername</i> in the current session. dscp prompts you for information about <i>servername</i> . Press Return to accept the default value, which is shown in square brackets [ ]. Enter “#done” to exit modify mode.
open [ <i>dsname</i> ]	Opens a session for the specified directory service, where <i>dsname</i> is the directory service name. If you do not specify a value for <i>dsname</i> , this command opens a session for the default directory service. To open a session, specify the value “InterfacesDriver” for <i>dsname</i> .
quit	Exits dscp.
read <i>servername</i>	Displays the contents of server entry <i>servername</i> .
sess	Lists all open sessions.
[switch] sess	Makes session number <i>sess</i> the current session.

---



# Utility Commands Reference

This chapter contains reference pages for the Adaptive Server utility program commands.

<b>Topic</b>	<b>Page</b>
Getting started	116
Utilities quick reference	118
backupserver	122
bcp	128
buildmaster	142
certauth	143
certpk12	147
certreq	150
charset	153
cobpre	154
cpre	155
dataserver	156
ddlgen	163
defncopy	186
dscp	192
dsedit	193
extractjava	194
installjava	197
isql	201
langinstall	213
optdiag	216
preupgrade	223
pwdcrypt	228
showserver	229
sqldbgr	230
sqlloc	235
sqllocres	236
sqlsrvr	237
sqlupgrade	244

Topic	Page
sqlupgraderes	245
srvbuild	246
srvbuildres	248
startserver	249
sybmigrate	252
xpserver	257

## Getting started

UNIX

You enter a utility program command at the system prompt in a UNIX shell.

Windows 2000, 2003,  
and XP

If a utility:

- Has an icon in the Sybase for Windows or Sybase for the Windows program group, double-click the icon to launch the utility program.
- Does not have an icon in the program group, enter the utility program command at the Windows command prompt to launch the utility program.

Place characters with special meaning to the shell (the command prompt in Windows), such as the backslash (\), asterisk (\*), slash (/), and spaces, in quotes. You can precede some special characters with the backslash (\) to “escape” them. This prevents the shell (command prompt) from interpreting the special characters.

Table 7-1 describes the utility programs available with Adaptive Server.

---

**Note** The utility programs described in Table 7-1 may allow you to use a -P parameter to enter your password. If security is an issue, do not use this parameter to specify your password. Another user may have an opportunity to see it. Instead, log in as usual without the -P parameter, and let Adaptive Server prompt you for your password.

---

**Table 7-1: Utility programs**

Utility	Description
backupserver	Executable form of the Backup Server™ program.
bcp	Copies rows in a database table to or from an operating system file in a user-specified format.
certauth	Converts a server-certificate request into a certificate authority-signed certificate.
certpk12	Export or import a PKCS#12 file.

<b>Utility</b>	<b>Description</b>
certreq	Creates a server certificate request and corresponding private key in two ways:
charset	Loads the character sets and sort order files.
cobpre	Precompiler for COBOL.
cpre	Precompiler for C.
dataserver	Executable form of the Adaptive Server program.
ddlgen	Generates data definition language for server- and database-level objects in ASE.
defncopy	Copies definitions for specified views, rules, defaults, triggers, procedures, or reports from a database to an operating system file or from an operating system file to a database.
dscp	Allows you to view and edit server entries in the interfaces file in command-line mode.
dsedit	Allows you to view and edit server entries in the interfaces file using a graphical user interface based on X11/Motif.
extractjava	Copies a retained JAR from an Adaptive Server to a client file.
installjava	Installs a JAR from a client file into an Adaptive Server.
isql	Interactive SQL parser to Adaptive Server.
langinstall	Installs a new language on the Adaptive Server.
optdiag	Displays optimizer statistics or loads updated statistics into system tables.
preupgrade	Performs tests on an installation or database to determine its readiness for upgrade.
pwdcrypt	Creates and prints an encrypted LDAP password in the libtcl.cfg file.
showserver	Shows Adaptive Servers and Backup Servers that are currently running on the local machine.
sqldbg	Debugs stored procedures and triggers.
sqlloc	Installs and modifies languages, character sets, and sort order defaults for Adaptive Server in GUI mode.
sqllocres	Installs and modifies languages, character sets, and sort order defaults for Adaptive Server in command-line mode.
sqlsrvr	Executable form of the Adaptive Server program.
sqlupgrade	Upgrades your currently installed release of Adaptive Server to the newest release in GUI mode.
sqlupgraderes	Upgrades your currently installed release of Adaptive Server to the newest release in command-line mode.
srvbuild	Creates a new Adaptive Server, Backup Server, Monitor Server, or XP Server in GUI mode with default or user-specified values for key configuration attributes.
srvbuildres	Creates a new Adaptive Server, Backup Server, Monitor Server, or XP Server in command-line mode with default or user-specified values for key configuration attributes.
sqlsrvr	Executable form of the Adaptive Server program.
startserver	Starts an Adaptive Server or a Backup Server.
srvmgr	Starts Adaptive Server and Backup Server as Windows services.
sybload	Uploads Sybase products from the distribution media and builds the Sybase installation directory from the command line.

Utility	Description
sybmigrate	Enables you to migrate database from a server using 2K logical pages to a server using 4, 8, or 16K logical pages.
sybsetup	Installs and configures Adaptive Server from a single location using a GUI interface.
xpserver	Starts XP Server manually.

## \*\_r utilities

Sybase provides you with the \_r versions of some of the utilities for use with threaded drivers.

The utilities in this manual that have \_r versions are:

- bcp
- cobpre
- cpre
- defncopy
- dscp
- isql

## Utilities quick reference

This section provides a quick reference for the utilities, divided into the following categories:

- “Installation or configuration utilities” on page 119
- “Utilities for languages, character sets, and sort orders” on page 119
- “Utilities to start servers” on page 120
- “Database creation and manipulation utilities” on page 120
- “Utilities to gather information” on page 121



## Installation or configuration utilities

Use the following to install or configure databases:

dataserver	Allows you to build a new Adaptive Server.
dscp	Allows you to view and edit server entries in the interfaces file from the command line.
dsedit	Allows you to view and edit server entries in the interfaces file using a GUI based on X11/Motif in UNIX platforms.  In Windows, allows you to create and modify network connection information in the interfaces file.
preupgrade	Performs tests on an installation or database to determine its readiness for upgrade, and reports problems found.
sqlupgrade	Upgrades your currently installed release of Adaptive Server to the newest release using a GUI based on X11/Motif in UNIX platforms.
sqlupgraderes	Upgrades your currently installed release of Adaptive Server to the newest release using resource files in UNIX platforms.
srvbuild	Creates a new Adaptive Server, Backup Server, Monitor Server, or XP Server with default or user-specified values for key configuration attributes using a graphical user interface based on X11/Motif in UNIX platforms.
srvbuildres	Creates a new Adaptive Server, Backup Server, Monitor Server, or XP Server, using resource files to specify values for key configuration attributes in UNIX platforms.

## Utilities for languages, character sets, and sort orders

Use the following utilities to set languages, character sets and sort orders:

charset	Loads the character sets and sort order files in Windows.
langinstall	Installs a new language on an Adaptive Server.
sqlloc	Installs and modifies languages, character sets, and sort order defaults for Adaptive Server, using a GUI based on X11/Motif in UNIX platforms.
sqllocres	Installs and modifies languages, character sets, and sort order defaults for Adaptive Server, using a resource file in UNIX platforms.

## Utilities to start servers

Use the following utilities to start servers manually:

backupserver	Starts the Backup Server executable. Use the startserver command instead of this utility to start Backup Server manually. In Windows, you can use the srvmgr utility instead to start Backup Server manually.
dataserver	Starts the Adaptive Server executable. Use the startserver command instead of this utility to start Adaptive Server manually.
histserver	Starts the Historical Server executable. Use the histserver command instead of this utility to start Historical Server manually.
monserver	Starts the Monitor Server executable. Use the monserver command instead of this utility to start Monitor Server manually.
sqlsrv	Starts the Adaptive Server executable in Windows . Use the services manager utility instead of this utility to start Adaptive Server manually.
srvmgr	Starts, pauses, and stops Adaptive Server, Backup Server, and Adaptive Server Monitor™ as Windows services.
startserver	Starts an Adaptive Server and a Backup Server in UNIX platforms.

## Database creation and manipulation utilities

Use the following utilities to create and manipulate databases:

bcp	Copies a database table to or from an operating system file in a user-specified format.
defncopy	Copies definitions for specified views, rules, defaults, triggers, or procedures from a database to an operating system file or from an operating system file to a database.
extractjava	Copies a retained JAR and the classes it contains from an Adaptive Server to a client file.
installjava	Installs a JAR from a client file into an Adaptive Server database.
isql	Interactive SQL parser to Adaptive Server.
optdiag	Displays optimizer statistics or loads updated statistics into system table.

## Utilities to gather information

Use the following utilities to gather information:

- |            |  |
|------------|--|
| showserver | Shows the Adaptive Servers and Backup Servers that are currently running on the local machine in UNIX platforms. |
| wdllvers   | Provides information about the Sybase DLLs (dynamic link libraries) that are loaded into memory in Windows.      |

## backupserver

Description	<p>The executable form of the Backup Server program, located in <code>\$\$SYBASE/\$\$SYBASE_ASE/bin</code>.</p> <p><b>Windows</b> The utility is <code>bcksvr.exe</code>, located in <code>\$\$SYBASE%/\$\$SYBASE_ASE%/bin</code>.</p>
Syntax	<pre> backupserver     [-C server_connections]     [-S b_servername]     [-I interfaces_file]     [-e error_log_file]     [-M sybmultipbuf_binary]     [-N network_connections]     [-T trace_value]     [-L Sybase_language_name]     [-J Sybase_character_set_name]     [-c tape_config_file]     [-D n]     [-A pathname]     [-P active_service_threads]     [-V level_number]     [-p n]     [-m max_shared_memory]         </pre> <p>Or</p> <pre> backupserver -v         </pre>
Parameters	<p><b>-C <i>server_connections</i></b>          specifies the number of server connections for the Backup Server. The Backup Server requires:</p> <ul style="list-style-type: none"> <li>• Two connections for each dump session</li> <li>• One connection for each load session</li> <li>• One connection for volume change messages</li> </ul> <p>Allow a maximum of three times the number of expected concurrent dump and load sessions. The default value is 30 server connections.</p> <p><b>-S <i>b_servername</i></b>          specifies the name of the Backup Server to start. The default is <code>SYB_BACKUP</code>. This entry must specify the name of a Backup Server in the interfaces file.</p>

- l *interfaces\_file*  
specifies the name and location of the interfaces file to search when connecting to Backup Server. If -l is omitted, backupserver looks for a file named *interfaces* in the directory pointed to by your SYBASE environment variable.
- v  
prints the version number and copyright message of the backupserver software and then exits.
- e *error\_log\_file*  
specifies the name and location of the Backup Server error log file used to report Open Server internal errors, sybmultbuf errors, errors that halt the Backup Server, and errors for disconnected sessions. All other errors are sent to the notify destination specified in the dump database, dump transaction, load database, and load transaction commands.
- M *sybmultbuf\_binary*  
specifies the full path name of the sybmultbuf executable. Use this parameter only when starting Backup Server from a directory other than the *bin* directory of the Sybase installation directory, or when using a diagnostic version of sybmultbuf.
- N *network\_connections*  
specifies the number of total network connections (DBPROCESSes) that the master Backup Server can originate. The default value is 25.
- T *trace\_value*  
interprets *trace\_value* as a bitmask (base-2 number). The 1 bits in *trace\_value* correspond to Open Server Trace flags to turn on. If you specify more than one -T parameter on the command line, the final -T value overrides the values from earlier -T parameters. The *trace\_value* must be a positive integer.
- L *Sybase\_language\_name*  
specifies the default language for Backup Server. If not specified, Backup Server uses the locale specified by the LC\_ALL or LANG environment variables. If these variables are not set, Backup Server searches for the “default” entry in *locales.dat*.

---

**Note** The -L parameter does not override the value set in the LANG environment variable.

---

- J *Sybase\_character\_set\_name*  
specifies the default character set for Backup Server.

- c *tape\_config\_file*  
specifies the name and location of the tape configuration file to search for tape device configuration information before doing a dump database or a dump transaction. If you do not specify -c, the default path name for the tape configuration file is *\$SYBASE/backup\_tape.cfg*.
- D *n*  
specifies the bitmap (base 10 number) of the diagnostic flags used within Backup Server.
- A *pathname*  
specifies the pathname to the directory of the Archive API dynamically loadable library.
- P *active\_service\_threads*  
allows you to increase the number of stripes during multiple dump/load operations (with a maximum of 12286 stripes per single operation).

**-V *level\_number***

limits the messages that are printed to the Backup Server error log. The *level\_number* variable determines the degree of error verbosity (-V) for Backup Server:

- -V3 – displays only completion messages from a normal dump or load command and the following types of messages:
  - Error messages from Backup Server and sybmultbuf
  - Other sybmultbuf messages
  - Volume change messages
  - Open Server™ messages
  - Trace print messages
  - Informational messages from the System & Tape Auto Config modules
- -V2 – displays:
  - All -V3 messages **plus**
  - File creation and file mount messages
- -V1 – displays:
  - All -V2 messages **plus**
  - Phase messages
- -V0 (default) – displays all messages, including backup progress

This limitation does not involve the messages that are sent to the client or console as determined by the NOTIFY= parameter in a dump or load command.

This option also does not affect logging for the following message types:

- Open Server messages
- Trace printing messages from bs\_traceprint
- sybmultbuf messages

-p *n*

specifies the TDS packet size in bytes that the **local** Backup Server requests from the **remote** Backup Server during network dumps. The actual packet size used is limited to the -p parameter value of the remote Backup Server. If you do not specify -p, the default is 2048 bytes. The packet size should be an integer greater than, or equal to 256.

-m *max\_shared\_memory*

specifies the maximum amount of shared memory in megabytes that Backup Server can use for all of its dump or load sessions.

Usage

- Start Backup Server with the startserver command rather than by directly executing the backupserver program.
  - To change default values in UNIX, edit the *RUN\_servername* file in your Sybase installation directory. See the startserver reference page for details.
  - To change default values in Windows, use Server Config to change the command-line parameters of the Backup Server. See the *Configuration Guide* for details.
- Make sure that the device driver options you include with the dump command are accurate. Backupserver does not verify any device driver options you include during a dump command. For example, if you include an option that forces Backupserver to rewind a tape before use, it will always rewind the tape to the beginning instead of reading the tape from the point of the dump.
- If you do not specify a Backup Server name with the -S parameter, and you have not set the environment variable DSLISTEN, backupserver uses the default Backup Server name SYB\_BACKUP in UNIX.
 

**In Windows** – bcksvr uses the default Backup Server name *server\_name\_BS*. The value of the DSLISTEN environment variable overrides this default value, and the -S parameter overrides both the default and the value specified in DSLISTEN.
- Whenever possible, the Backup Server and any Adaptive Servers that dump or load directly through the Backup Server should share the same interfaces file (*sql.ini* in UNIX). The interfaces file that Backup Server uses must contain entries for:
  - Backup Server
  - Any other Backup Servers with which this Backup Server communicates



- Trace flags cause the Backup Server to print information regarding its operation while it is running, for debugging problems in the Backup Server. See the *Open Server Server-Library/C Reference Manual* for more details on trace flags. Backup Server does not support use of the Open Server-defined SRV\_TR symbols for -T.
- If Backup Server cannot find the *locales* and *charsets* directories specified by the -L and -J parameters, or if these parameters specify an incorrect language and character set combination, Backup Server issues an error message and uses the default language and character set.
- Backup Server cannot perform loads or dumps between servers that use different logical page sizes. For example, you can load a 4K logical page sized database dump into another server using a 4K logical page size. But Backup Server does not support dumping a 4K logical page sized database and loading it into a database that uses 16K logical page size.

**Permissions**

Anyone with execute permission on the binary, and who has read/write access to all the files.

**See also**

**Utilities** startserver

## bcp

**Description** Copies a database table to or from an operating system file in a user-specified format. bcp is located in `$$SYBASE/$SYBASE_OCS/bin`.

**Windows** The utility is `bcp.exe`, and is located in `%%SYBASE%\%%SYBASE_OCS%\bin`.

**Syntax**

```
bcp [[database_name.]owner.]table_name [: [ partition_id | slice_number ] |  
partition partition_name] {in | out} datafile  
[--show-fi]  
[--hide-vcc]  
[-f formatfile]  
[-e errfile]  
[-F firstrow]  
[-L lastrow]  
[-b batchsize]  
[-m maxerrors]  
[-n]  
[-c]  
[-t field_terminator]  
[-r row_terminator]  
[-U username]  
[-P password]  
[-l interfaces_file]  
[-S server]  
[-a display_charset]  
[-z language]  
[-A packet_size]  
[-J client_charset]  
[-T text_or_image_size]  
[-E]  
[-g id_start_value]  
[-N]  
[-X]  
[-K keytab_file]  
[-R remote_server_principal]  
[-V [security_options]]  
[-Z security_mechanism]  
[-Q]  
[-Y]  
[--maxconn maximum_connections]  
[--initstring T-SQL command]
```

Or

```
bcp -v
```

**Parameters** `maxconn maximum_connections` is the maximum number of parallel connections permitted for each bulk copy operation. The default is 10.

`--show-fi`

instructs `bcp` to copy functional indexes, while using either `bcp in` or `bcp out`.

Using this option sends the data from a functional index to or from the server.

`--hide-vcc`

instructs `bcp` not to copy virtual computed columns either to or from a datafile. When you use this option in `bcp OUT`, the data file contains no data for virtual computed columns. When you use it in `bcp IN`, the data file may contain no data for a virtual computed column.

If you use this option, Adaptive Server does not calculate or send virtual computed column data.

*database\_name*

is optional if the table being copied is in your default database or in master. Otherwise, you must specify a database name.

*owner*

is optional if you or the Database Owner owns the table being copied. If you do not specify an owner, `bcp` looks first for a table of that name that you own, and then looks for one owned by the Database Owner. If another user owns the table, you must specify the owner name or the command fails.

*table\_name*

is the name of the database table to copy. The table name cannot be a Transact-SQL reserved word.

*partition\_id*

specifies the partition number into which data is to be copied. It is supported only for `bcp in`. It is the equivalent of `slice_number` in Adaptive Server 12.5.x.

*slice\_number*

specifies the partition slice into which data is to be copied. It is supported only for `bcp in` and only for round-robin partitioned tables in Adaptive Server 15.0 and later.

partition *partition\_name*

specifies a set of one or more partitions, separated by commas.

`in | out`

is the direction of the copy. `in` indicates a copy from a file into the database table; `out` indicates a copy to a file from the database table or view.

*datafile*

specifies a set of one or more unique data files, separated by commas. It is supported for both *bcp in* and *bcp out*. The path name can be from 1 to 255 characters in length.

*-f formatfile*

is the full path name of a file with stored responses from a previous use of *bcp* on the same table. After you answer *bcp*'s format questions, it prompts you to save your answers in a format file. Creation of the format file is optional. The default file name is *bcp.fmt*. The *bcp* program can refer to a format file when you are copying data so that you do not have to duplicate your previous format responses interactively. Use the *-f* parameter only if you previously created a format file that you want to use now for a copy in or copy out. If you do not specify this parameter, *bcp* interactively queries you for format information.

*-e errfile*

is the full path name of an error file where *bcp* stores any rows that it was unable to transfer from the file to the database. Error messages from *bcp* appear on your terminal. *bcp* creates an error file only when you specify this parameter.

*-F firstrow*

is the number of the first row to copy from an input file (default is the first row).

Avoid using the *-F* option when performing heavy-duty, multi-process copying, as it causes *bcp* to generally spend more effort to run, and does not provide you with a faster process. Instead, use *-F* for single-process, ad-hoc copying.

*-L lastrow*

is the number of the last row to copy from an input file (default is the last row).

**-b** *batchsize*

is the number of rows per batch of data copied. By default, bcp in copies *n* rows in one batch, where *n* is equal to the batch size. Batching applies only when you are bulk copying in; it has no effect on bulk copying out. The smallest number bcp accepts for *batchsize* is 1.

---

**Note** Setting the batch size to 1 causes Adaptive Server to allocate one data page to one row copied in. This option only applies to fast bcp, and is only useful in locating corrupt rows of data. Use -b1 with care—doing so causes a new page to be allocated for each row, and is a poor use of space.

---

**-m** *maxerrors*

is the maximum number of nonfatal errors permitted before bcp aborts the copy. bcp discards each row that it cannot insert (due to a data conversion error, or an attempt to insert a null value into a column that does not allow them), counting each rejected row as one error. If you do not include this parameter, bcp uses a default value of 10.

**-n**

performs the copy operation using native (operating system) formats. Specifying the -n parameter means bcp will not prompt for each field. Files in native data format are not human-readable.

---

**Warning!** Do not use:

- bcp in native format for data recovery or salvage or to resolve an emergency situation.
- bcp in native format to transport data between different hardware platforms, different operating systems, or different major releases of Adaptive Server.
- field terminators (-t) or row terminators (-r) with bcp in native format.

Results are unpredictable and data may become corrupted.

Using bcp in native format can create flat files that cannot be reloaded into Adaptive Server and it may be impossible to recover the data. If you cannot rerun bcp in character format (for example, a table was truncated or dropped, hardware damage occurred, a database was dropped, and so on) the data is unrecoverable.

---

- c  
performs the copy operation with char datatype as the default storage type of all columns in the data file. Use this format if you are sharing data between platforms. This parameter does not prompt for each field; it uses char as the default storage type, no prefixes, \t (tab) as the default field terminator, and \n (new line) as the default row terminator.
- t *field\_terminator*  
specifies the default field terminator.
- r *row\_terminator*  
specifies the row terminator.

---

**Warning!** Do not use -t or -r parameters with bcp in native format. Results are unpredictable and data may become corrupted.

---

When specifying terminators from the command line with the -t or -r parameter, you must escape characters that have special significance to the UNIX operating system (or the command prompt shell for Windows). See the examples for bcp for more information. Either place a backslash in front of the special character or enclose it in quotes. This is not necessary when bcp prompts you (interactive mode).

- U *username*  
specifies an Adaptive Server login name.
- P *password*  
specifies an Adaptive Server password. If you do not specify -P*password*, bcp prompts for a password. You can leave out the -P flag if your password is NULL.
- I *interfaces\_file*  
specifies the name and location of the interfaces file to search when connecting to Adaptive Server. If you do not specify -I, bcp looks for an interfaces file (*sql.ini* in Windows) located in the directory specified by the SYBASE environment variable (*ini* directory in Windows).
- S *server*  
specifies the name of the Adaptive Server to which to connect. If you specify -S with no argument, bcp uses the server specified by the DSQUERY environment variable.

**-a** *display\_charset*

allows you to run `bcp` from a terminal where the character set differs from that of the machine on which `bcp` is running. Use `-a` in conjunction with `-J` to specify the character set translation file (*.xlt* file) required for the conversion. Use `-a` without `-J` only if the client character set is the same as the default character set.

The following error message appears if the character translation file(s) named with the `-a` parameter is missing, or you mistype the name(s):

```
Error in attempting to determine the size of a pair of
translation tables.: 'stat' utility failed.
```

**-z** *language*

is the official name of an alternate language the server uses to display `bcp` prompts and messages. Without the `-z` flag, `bcp` uses the server's default language.

You can add languages to an Adaptive Server during installation or afterwards, using either the `langinstall` utility (or `langinst` in Windows) or the `sp_addlanguage` stored procedure.

The following error message appears if an incorrect or unrecognized language is named with the `-z` parameter:

```
Unrecognized localization object. Using default value
'us_english'.
Starting copy...
=> warning.
```

**-v**

displays the version number of `bcp` and a copyright message and returns to the operating system.

**-A** *packet\_size*

specifies the network packet size to use for this `bcp` session. For example, the following sets the packet size to 4096 bytes for this `bcp` session:

```
bcp pubs2..titles out table_out -A 4096
```

*packet\_size* must be between the values of the default network packet size and maximum network packet size configuration variables, and it must be a multiple of 512.

Use network packet sizes larger than the default to improve the performance of large bulk-copy operations.

**-J *client\_charset***

specifies the character set to use on the client. bcp uses a filter to convert input between *client\_charset* and the Adaptive Server character set.

-J *client\_charset* requests that Adaptive Server convert to and from *client\_charset*, the character set used on the client.

-J with no argument sets character set conversion to NULL. No conversion takes place. Use this if the client and server use the same character set.

Omitting -J sets the character set to a default for the platform, which may not necessarily be the character set that the client is using.

This error message appears if an incorrect or unrecognized character set is named with the -J parameter:

```
Unrecognized localization object. Using default value
'iso_1'.
Starting copy...
=> warning.
```

For more information about character sets and associated flags, see the *System Administration Guide*.

**-T *text\_or\_image\_size***

allows you to specify, in bytes, the maximum length of text or image data that Adaptive Server sends. The default is 32K. If a text or an image field is larger than the value of -T or the default, bcp does not send the overflow.

**-E**

explicitly specifies the value of a table's IDENTITY column.

By default, when you bulk copy data into a table with an IDENTITY column, bcp assigns each row a temporary IDENTITY column value of 0. This is effective only when copying data into a table. bcp reads the value of the ID column from the data file, but does not send it to the server. Instead, as bcp inserts each row into the table, the server assigns the row a unique, sequential, IDENTITY column value, beginning with the value 1. If you specify the -E flag when copying data into a table, bcp reads the value from the data file and sends it to the server which inserts the value into the table. If the number of rows inserted exceeds the maximum possible IDENTITY column value, Adaptive Server returns an error.

The -E parameter has no effect when you are bulk copying data out. Adaptive Server copies the ID column to the data file, unless you use the -N parameter.

You cannot use the -E and -g flags together.



-g *id\_start\_value*

specifies the value of the IDENTITY column to use as a starting point for copying data in.

You cannot use the -g and -E flags together.

-N

skips the IDENTITY column. Use this parameter when copying data in if your host data file does not include a placeholder for the IDENTITY column values, or when copying data out, if you do not want to include the IDENTITY column information in the host file.

You cannot use both -N and -E parameters when copying data in.

-X

specifies that, in this connection to the server, the application initiates the login with client-side password encryption. bcp (the client) specifies to the server that password encryption is desired. The server sends back an encryption key, which bcp uses to encrypt your password, and the server uses the key to authenticate your password when it arrives.

If bcp crashes, the system creates a core file that contains your password. If you did not use the encryption option, the password appears in plain text in the file. If you used the encryption option, your password is not readable.

-K *keytab\_file*

specifies the path to the keytab file used for authentication in DCE.

-R *remote\_server\_principal*

specifies the principal name for the server as defined to the security mechanism. By default, a server's principal name matches the server's network name (which is specified with the -S parameter or the DSQUERY environment variable). Use the -R parameter when the server's principal name and network name are not the same.

**-V** *security\_options*

specifies network-based user authentication. With this option, the user must log in to the network's security system before running the utility. In this case, users must supply their network user name with the -U option; any password supplied with the -P option is ignored.

You can follow -V with a *security\_options* string of key-letter options to enable additional security services. These key letters are:

- c – enables data confidentiality service
- i – enables data integrity service
- m – enables mutual authentication for connection establishment
- o – enables data origin stamping service
- r – enables data replay detection
- q – enables out-of-sequence detection

**-Z** *security\_mechanism*

specifies the name of a security mechanism to use on the connection.

Security mechanism names are defined in the *\$\$SYBASE/install/libtcl.cfg* configuration file. If no *security\_mechanism* name is supplied, the default mechanism is used. For more information on security mechanism names, see the description of the *libtcl.cfg* file in the *Open Client and Open Server Configuration Guide*.

**-Q**

provides backward compatibility with bcp version 10.0.4 for copying operations involving nullable columns.

**-Y**

specifies that character-set conversion is disabled in the server, and is instead performed by bcp on the client side when using bcp in.

---

**Note** A client-side Unicode conversion is supported only for Adaptive Server 15.0 and later.

All character-set conversion is done in the server during bcp out.

---

**--maxconn** *maximum\_connections*

is the maximum number of parallel connections that the bcp client can open to the server. If --maxconn is not specified, bcp determines the number of connections to use.

--initstring *T-SQL command*

allows you to send a Transact-SQL command through bcp just after the connection to Adaptive Server is established but before the actual data transfer starts.

## Examples

**Example 1** Copies data from the publishers table to a file named *pub\_out* with char datatype as the default storage type of all columns in the data file, specifying the default field terminator and row terminator.

**In UNIX platforms** – The first backslash before the final “r” escapes the second so that only one backslash is printed:

```
bcp pubs2..publishers out pub_out -c -t , -r \\r
```

**In Windows:**

```
bcp pubs2..publishers out pub_out -c -t , -r \r
```

**Example 2** Copies data from the publishers table to a file named *pub\_out* for later reloading into Adaptive Server. Press Return to accept the defaults specified by the prompts. The same prompts appear when you copy data into the publishers table:

```
bcp pubs2..publishers out pub_out
Password:
Enter the file storage type of field pub_id [char]:
Enter prefix length of field pub_id [0]:
Enter length of field pub_id [4]:
Enter field terminator [none]:
Enter the file storage type of field pub_name [char]:
Enter prefix length of field pub_name [1]:
Enter length of field pub_name [40]:
Enter field terminator [none]:
Enter the file storage type of field city [char]:
Enter prefix length of field city [1]:
Enter length of field city [20]:
Enter field terminator [none]:
Enter the file storage type of field state [char]:
Enter prefix length of field state [1]:
Enter length of field state [2]:
Enter field terminator [none]:
```

**In UNIX, you are then asked:**

```
Do you want to save this format information in a
file? [Y-n] y
Host filename [bcp.fmt]: pub_form
Starting copy...
3 rows copied.
```

```
Clock Time (ms.): total = 1 Avg = 0 (3000.00 rows per
sec.)
```

**Example 3** Copies data back into Adaptive Server using the saved format file, *pub\_form*:

```
bcp pubs2..publishers in pub_out -f pub_form
```

**Example 4** Enter the single letter exactly as it appears below:

To see examples of datatypes, enter "?" at the prompt:

```
Enter the file storage type of field 'pub_id'
```

```
['char']:?
```

```
Invalid column type. Valid types are:
```

```
<cr>: same type as Adaptive Server column.
```

```
c : char
```

```
T : text
```

```
i : int
```

```
s : smallint
```

```
t : tinyint
```

```
f : float
```

```
m : money
```

```
b : bit
```

```
d : datetime
```

```
x : binary
```

```
I : image
```

```
D : smalldatetime
```

```
r : real
```

```
M : smallmoney
```

```
n : numeric
```

```
e : decimal
```

**Example 5** Copies a data file created with a character set used on a VT200 terminal into the pubs2..publishers table. The -z flag displays bcp messages in French:

```
bcp pubs2..publishers in vt200_data -J iso_1 -z french
```

**Example 6 UNIX platforms only** – Specifies that you are using a Macintosh, running bcp on a workstation that is using roman8:

```
bcp pubs2..publishers in -a mac -J roman8
```

**Example 7** Specifies that Adaptive Server send 40K of text or image data using a packet size of 4096 bytes:

```
bcp pubs2..publishers out -T 40960 -A 4096
```

**Example 8** Sets 2 as the maximum number of parallel connections permitted for each operation.

```
bcp_r --maxconn 2
```

**Example 9** Copies the *mypart.dat* file of the current directory to partition p1 of table t1.

```
bcp t1 partition p1 in mypart.dat
```

**Example 10** Copies partition p1, p2, and p3 to files *a*, *b*, and *c* respectively, into the `\work2\data` directory.

```
bcp t1 partition p1, p2, p3 out \work2\data\1,
\work2\data\b, \work2\data\c
```

**Example 11** Disables replication for the bcp connection during the transfer of data from *titles.txt* data into *pubs2..titles*:

```
bcp pubs2..titles in titles.txt -initstring 'set
replication off'
```

**Example 12** Copies out database *db\_1*, which includes table t1 with materialized computed column c1:

```
bcp db_1..t1 out db_1.dat -Usa -P -S big_db -I./interfaces -f ./bcp.fmt
```

**Example 13** Copies in the data file (*db\_1.dat*) created in Example 12, containing table t1 with materialized computed column c1:

```
bcp db_1..t1 in db_1.dat -Usa -P -S big_db -I./interfaces -f ./bcp.fmt
```

#### Usage

- Use this syntax for bcp if you are using threaded drivers.
- You cannot use named pipes to copy files in or out.
- Error message format is different than earlier versions of bcp. If you have scripts that perform routines based on the values of these messages you may need to rewrite them, for example:

```
The display message that indicates the number of rows
transferred has been changed. During a session, this
version of bcp periodically reports a running total
of rows transferred. This message replaces the "1000
rows transferred" message displayed by the previous
bcp.
```

- Using `--hide-vc` improves performance, as Adaptive Server does not transfer and calculate data from virtual computed columns.
- `slice_number` is included for backward compatibility with Adaptive Server 12.5.x and earlier, and can be used only with round-robin partitioned tables.
- You can specify either `partition_id` or `partition_name`, but not both.

- You can specify multiple partitions and data files. Separate each partition name or data file name with commas.
- If you provide no partition name, bcp copies to the entire table.
- When using bcp out:
  - If *partition\_name* and *datafile* are both specified, then either *datafile* must specify a single data file, or you must specify a one-to-one mapping between partition names and data files.
  - If *datafile* is not specified, data from each partition is copied to a file named for the named partition with a *.dat* extension. For example, if the partition name is *ptn1*, the data file is *ptn1.dat*.
- You can use *initstring* to run any Transact-SQL command, but you must reset any permanent changes to the server *initstring* causes after bcp finishes. For instance, in Example 11 above, if the Adaptive Server account doesn't have the appropriate permissions, Adaptive Server returns an error message for the initialization string. bcp displays the server error message and stops before any data is transferred.

Result sets issued by the initialization string are silently ignored unless an error occurs.

- When using bcp in, if *partition\_name* is specified, *datafile* must specify a corresponding number of data files

#### Permissions

You must have an Adaptive Server account and the appropriate permissions on the database tables or views, as well as the operating system files to use in the transfer to use bcp.

- To copy data into a table, you must have insert permission on the table.
- To copy a table to an operating system file, you must have select permission on the following tables:
  - The table to copy
  - sysobjects
  - syscolumns
  - sysindexes

#### Auditing

Values in event and extrainfo columns are:

Event	Audit option	Command or access audited	Information in extrainfo
4	bcp	bcp in	<ul style="list-style-type: none"> <li>• <i>Roles</i> – Current active roles</li> <li>• <i>Keywords or options</i> – NULL</li> <li>• <i>Previous value</i> – NULL</li> <li>• <i>Current value</i> – NULL</li> <li>• <i>Other information</i> – NULL</li> <li>• <i>Proxy information</i> – Original login name, if set proxy in effect</li> </ul>

Tables used sysaudits\_01 – sysaudits\_08

See also **Commands** insert

**Documentation** Chapter 4, “Using bcp to Transfer Data to and from Adaptive Server” for an in-depth discussion of bcp, and the *Performance and Tuning Guide* for more information on how changing certain parameters can affect bcp for large batches.

**System procedures** sp\_audit, sp\_dboption, sp\_displayaudit

## **buildmaster**

Description	Adaptive Server version 12.5 and later no longer uses the buildmaster binary to build the master device. Instead, Sybase has incorporated the buildmaster functionality in the dataserver binary. See Chapter 1, “Building Servers Using dataserver” for more information, and dataserver on page 156 for syntax.
Syntax	None.



## certauth

Description	<p>Converts a server certificate request to a CA- (certificate authority) signed certificate. Located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code>.</p> <p><b>Windows</b> The utility is <code>certauth.exe</code>, and is located in <code>\$\$SYBASE%\\$\$SYBASE_OCS%\bin</code>.</p>
Syntax	<pre>certauth     [-r]     [-C <i>caCert_file</i>]     [-Q <i>request_filename</i>]     [-K <i>caKey_filename</i>]     [-N <i>serial_number</i>]     [-O <i>SignedCert_filename</i>]     [-P <i>caPassword</i>]     [-s <i>start_time</i>]     [-T <i>valid_time</i>]  Or  certauth -v</pre>
Parameters	<p><code>-r</code> when specified, creates a self-signed root certificate for the test environment.</p> <p><code>-C <i>caCert_file</i></code> specifies the name of the CA's certificate request file when <code>-r</code> is specified, or specifies the name of the CA's root certificate.</p> <p><code>-Q <i>request_filename</i></code> specifies the name of certificate request file.</p> <p><code>-K <i>caKey_filename</i></code> specifies the name of the CA's private key.</p> <p><code>-N <i>serial_number</i></code> specifies the serial number in the signed certificate. If <code>-N</code> is not specified, <code>certauth</code> generates a pseudo-random serial number.</p> <p><code>-O <i>SignedCert_filename</i></code> specifies the name to use for the output when creating a signed certificate file. If <code>-r</code> is specified, <code>SignedCert_filename</code> is the self-signed root certificate. If <code>-r</code> option is not used, <code>SignedCert_filename</code> is the certificate signed by the <code>caCert_file</code>.</p> <p><code>-P <i>caPassword</i></code> specifies the CA's password that is used to decrypt its private key.</p>

- s *start\_time*  
specifies the start of the valid time range, measured in days from the current time. The default is the current time.
- T *valid\_time*  
specifies the length of the valid time range for a signed certificate. The valid time range is in units of days.
- v  
prints the version number and copyright message of the certauth tool, then exits.

### Examples

**Example 1** Converts the CA's certificate request (*ca\_req.txt*) to a certificate, using the private key (*ca\_pkey.txt*). The private key is protected using *password*. This example sets the valid time range to 365 days, self-signs the certificate, and outputs it as a root certificate (*trusted.txt*):

```
certauth -r -C ca_req.txt -Q ca_req.txt
        -K ca_pkey.txt -P password -T 365 -O trusted.txt
```

The utility returns this message:

```
-- Sybase Test Certificate Authority --
Certificate Validity:
  startDate = Tue Sep 5 10:34:43 2000
  endDate   = Wed Sep 5 10:34:43 2001
CA sign certificate SUCCEED (0)
```

---

**Note** You need to create a trusted root certificate for the test CA only once. After you have created the trusted root certificate, you can use it to sign many server certificates in your test environment.

---

**Example 2** Converts a server certificate request (*srv5\_req.txt*) to a certificate, and sets the valid time range to 180 days. It signs the certificate with a CA's certificate and private key (*trusted.txt* and *ca\_pkey.txt*), uses password protection, and outputs the signed certificate as *sybase\_srv5.crt*:

```
certauth -C trusted.txt -Q srv5_req.txt
        -K ca_pkey.txt -P password -T 180 -O sybase_srv5.crt
```

---

**Note** If you do not set valid time, the default is 365 days.

---

The utility returns this message:

```
-- Sybase Test Certificate Authority --
Certificate Validity:
```

```

      startDate = Tue Sep  5 10:38:32 2000
      endDate = Sun Mar  4 09:38:32 2001
      CA sign certificate SUCCEEDED (0)

```

The following is a sample certificate. See the `certauth` Usage section for additional steps to take to create a server certificate that the server can use.

```

-----BEGIN CERTIFICATE-----
MIICSTCCAgUCAVAwCwYHKOZIzjgEAWAMG8xCzAJBgNVBAYTA1VTMRMwEQYDVQOQI
EwpDYWxpZm9ybmlhMRRMwEQYDVQHEwpFbWVyeXZpbGx1MQ8wDQYDVQQKFAZTeWh
c2UxDDAKBgNVBAsUA0RTVDEXMBUGA1UEAxQOc3liYXNlX3R1c3RfY2EwHhcNMDAw
ODE4MTkxMzM0WhcNMDAwODE4MTkxMzM0WjBvMQswCQYDVQQGEwJVUzETMBEGAUE
CBMKQ2FsaWZvcn5pYTETMBEGA1UEBxMKRW11cn12aWxsZTEPMA0GA1UEChQGUGU3li
YXNlMQwwCgYDVQQQLFANEU1QxYFzAVBgNVBAMUDnN5YmFzZV90ZXN0X2NhMIHwMIo
Bgqhkhj00AQBMIGcAkEA+6xG7XCxik1xbP96nHBnQrTLTCjHlcy8QhIekwv9OlqG
EMG9AjJLxj6VCkPOD75vqVMEkaPPj0IbXEJEe/aYXQIVAPyvY1+B9phC2e2YFcf7
cReCcSNxAkBBHt7rnOJZ1Dnd8iLQgt0wd1w41o/Xx2OeZS4CJW0KVkGIId1hNGz8r
GrQTspWcwTh2rNgbXxlNXhAV5g4OCgrYA0MAAkA70uNE190KmhdT3RISiceCMgOf
1J8dgtWF15mchES8OmF9s/vqPAR5NkaV7LJK6k7QvXUBY+8LM0ugpJf/TYMASG
ByqGSM44BAMFAAMxADAuAhUAhM2Icn1pSavQtXFzXJUCoOmNlPkCFQDtE8RUGuo8

```

```

      DdxnQtPu9uJDmoBiUQ==

```

```

-----END CERTIFICATE-----

```

## Usage

- The maximum length of the serial number in the `-N` option is 20 hexadecimal characters. If the specified serial number is longer, `certauth` truncates the serial number to the maximum length.
- To create a server certificate file that Adaptive Server understands, append the certificate requestor's private key to the end of the signed certificate file. Using example 2 above, you would cut and paste `srv5_pkey.txt` to the end of the signed certificate file, `sybase_srv5.crt`.
- To create a trusted roots file that the server can load upon start-up, rename `trusted.txt` to `sybase_srv5.txt` where `sybase_srv5.txt` is the common name of the server.
- Then copy the `sybase_srv5.txt` file into the Adaptive Server installation directory, for example, `$$SYBASE/$$SYBASE_ASE/certificates`.
- The options `-s` and `-T` together specify the time range for the certificate.

The file, which is required for an SSL-based session, is used to start the SSL-enabled Adaptive Server.

After the CA's root certificate is created, you can use it to sign multiple server certificates.

See also

**Utilities** certpk12, certreq

## certpk12

Description	Export or import a PKCS #12 file into a certificates file and a private key. Located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code> .
	<b>Windows</b> The utility is <code>certpk12.exe</code> , and is located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code> .
Syntax	<pre>certpk12     {-O <i>Pkcs12_file</i>   -I <i>Pkcs12_file</i>}     [-C <i>Cert_file</i>]     [-K <i>Key_file</i>]     [-P <i>key_password</i>]     [-E <i>Pkcs12_password</i>]</pre>
	Or <code>certpk12 -v</code>
Parameters	<p><b>-O <i>Pkcs12_file</i></b> specifies the name of a PKCS #12 file to be exported. The file can contain a certificate plus a private key, a single certificate, or a single private key. Either <code>-O</code> or <code>-I</code> needs to be on.</p> <p><b>-I <i>Pkcs12_file</i></b> specifies the name of a PKCS #12 file to be imported. The file can contain a certificate plus a private key, a single certificate, or a single private key. Either <code>-I</code> or <code>-O</code> needs to be on.</p> <p><b>-C <i>Cert_file</i></b> specifies the name of certificate file to be exported to a PKCS #12 file if <code>-O</code> is on; or the name of certificate file to be imported from a PKCS #12 file if <code>-I</code> is on.</p> <p><b>-K <i>Key_file</i></b> specifies the name of private key file to be exported to a PKCS #12 file if <code>-O</code> is on; or the name of private key file to be imported from a PKCS #12 file if <code>-I</code> is on.</p> <p><b>-P <i>Key_password</i></b> specifies the password which is used to protect the private key specified by <code>-K</code>. If <code>-O</code> is on, the password is required to export the private key to a PKCS #12 file; if <code>-I</code> is on, the password is required to output the private key to a text file after it is imported from a PKCS #12 file.</p>

**-E *Pkcs12\_password***

specifies the password used to protect the PKCS #12 file. If **-O** is on, the password is used to encrypt the PKCS #12 file to be exported; if **-I** is on, the password is used to decrypt the PKCS #12 file to be imported. The password is also called “transport password.”

**-v**

prints the version number and copyright message of the certpk12 tool and exits.

## Examples

**Example 1** Exports *caRSA.crt*, the certificate file and *caRSAPkey.txt*, the private key file, to a PKCS#12 file (*caRSA.p12*). *password* is the password used to decrypt *caRSAPkey.txt*. *pk12password* is the password used to encrypt the final *caRSA.p12*:

```
certpk12 -O caRSA.p12 -C caRSA.crt -K caRSAPkey.txt
-P password -E pk12password
```

```
-- Sybase PKCS#12 Conversion Utility certpk12 Thu Nov 9 16:55:51 2000--
```

**Example 2** Imports *caRSA.p12*, a PKCS #12 file that contains a certificate and a private key. Output the embedded certificate to a text file (*caRSA\_new.crt*) and the embedded private key to a text file (*caRSAPkey\_new.txt*):

```
certpk12 -I caRSA.p12 -C caRSA_new.crt -K caRSAPkey_new.txt
-P new_password -E pk12password
```

```
-- Sybase PKCS#12 Conversion Utility certpk12 Thu Nov 9 16:55:51 2000--
```

*new\_password* is used to protect *caRSAPkey\_new.txt*, and *pk12password* is required to decrypt *caRSA.p12* file.

---

**Note** After you run examples 1 and 2, *caRSA.crt* and *caRSA\_new.crt* are identical. *caRSAPkey.txt* and *caRSAPkey\_new.txt* are different because they are encrypted randomly.

---

**Example 3** Exports the certificate file (*caRSA.crt*) to a PKCS#12 file (*caRSACert.p12*). *pkcs12password* is used to encrypt *caRSACert.p12*.

```
certpk12 -O caRSACert.p12 -C caRSA.crt -E pk12password
```

```
-- Sybase PKCS#12 Conversion Utility certpk12 Thu Nov 9 16:55:51 2000--
```

**Example 4** Imports a PKCS#12 file (*caRSACert.p12*) that contains a certificate. Output the embedded certificate to a text file (*caRSACert.txt*).

```
certpk12 -I caRSACert.p12 -C caRSACert.txt -E pk12password
```

```
-- Sybase PKCS#12 Conversion Utility certpk12 Thu Nov 9 16:55:51 2000--
```

*pk12password* is required to decrypt *caRSAcert.p12* file.

---

**Note** After you run Examples 3 and 4, the *caRSA.crt* and *caRSAcert.txt*, are identical.

---

Usage

- `certpk12` only supports triple-DES encrypted PKCS #12 file.
- Append certificate requestor's private key to the end of its signed certificate file.
- Name the file *servername.crt*, where *servername* is the name of the server. Place it in the certificates directory under `$SYBASE/$SYBASE_ASE` (`%SYBASE%\%SYBASE_ASE%` on Windows).

This file is needed to start the SSL-enabled Adaptive Server.

See also

**Utilities** `certauth`, `certreq`

## certreq

**Description** Creates a server certificate request and corresponding private key. This utility can be used in interactive mode, or you can provide all optional parameters on the command line. Located in `$SYBASE/$SYBASE_OCS/bin`.

**Windows** The utility is `certreq.exe`, and is located in `%SYBASE%\%SYBASE_OCS%\bin`.

**Syntax**

```
certreq
    [-F input_file]
    [-R request_filename]
    [-K PK_filename]
    [-P password]
```

Or

```
certreq -v
```

**Parameters**

**-F *input\_file***  
specifies the file name that contains attribute information to build a certificate request. If you do not specify an *input\_file* name, the required information must be interactively entered by a user.

The *input\_file* needs an entry for each of the following:

```
req_certtype={Server,Client}
req_keytype={RSA,DSA}
req_keylength={for RSA: 512-4096;
               for DSA: 512,768,1024}
req_country={string}
req_state={string}
req_locality={string}
req_organization={string}
req_orgunit={string}
req_commonname={string}
```

---

**Note** The common name must be the same as the server name.

---

See the Examples section for a sample file called *input\_file*.

**-R *request\_filename***  
specifies the name for the certificate-request file.

**-K *PK\_filename***  
specifies the name for the private-key file.

**-P *password***  
specifies the password used to protect the private key.



-v

displays the version number and copyright message, then exits.

#### Examples

**Example 1** This example does not use the `-F input_file` parameter, and is therefore in interactive mode. To create a server certificate request (*server\_req.txt*) and its private key (*server\_pkey.txt*), enter:

```
certreq
Choose certificate request type:
  S - Server certificate request
  C - Client certificate request (not supported)
  Q - Quit
Enter your request [Q] : s

Choose key type:

  R - RSA key pair
  D - DSA/DHE key pair
  Q - Quit

Enter your request [Q] : r

Enter key length (512, 768, 1024 for DSA; 512-2048 for
RSA) : 512

Country: US

State: california

Locality: dublin

Organization: sybase

Organizational Unit: dst

Common Name: server
```

The utility returns the message:

```
Generating key pair (please wait) . . .
```

After the key pair is generated, the `certreq` utility prompts you for more information.

```
Enter password for private key : password

Enter file path to save request: server_req.txt
```

Enter file path to save private key : *server\_pkey.txt*

**Example 2** In this sample text file, the format, *tag=value*, is used for noninteractive entry for a certificate request. You can use the -F option for noninteractive mode. When you use the -F option, be sure to use valid values and follow the format described above. Failure to do so prevents the certificate from being built correctly.

```
certreq -F input_file
req_certtype=server
req_keytype=RSA
req_keylength=512
req_country=us
req_state=california
req_locality=dublin
req_organization=sybase
req_orgunit=dst
req_commonname=server
```

After you create and save this file, enter on the command line, where *path\_and\_file* is the location of the text file:

```
certreq -F path_and_file -R server_req.txt -K server_pkey.txt -P password
```

This file creates a server certificate request, *server\_req.txt*, and its private key, *server\_pkey.txt* which is protected by *password*.

You can edit the server certificate file with any standard ASCII text editor.

#### Usage

- The input file uses the format of *tag=value*. *tag* is case-sensitive and should be the same as described above.
- The “=” is required. Valid *value* should start with a letter or digit, must be a single word, and there should not be any spaces within *value*.
- *value* is required for *req\_certtype*, *req\_keytype*, *req\_keylength* and *req\_commonname*.
- The space or tab around *<tag>*, = and *value* is allowed. Blank lines are also allowed.
- Each comment line should start with #.
- The certificate request file is in PKCS #10 format and used as acceptable input for the certauth tool to convert the request to a CA-signed certificate.

#### See also

**Utilities** certauth, certpk12

## charset

Description	<b>UNIX platforms only</b> Loads the character sets and sort order files in Adaptive Server. Located in <code>\$SYBASE/\$SYBASE_ASE/bin</code> .
Syntax	<pre>charset     [-P <i>password</i>]     [-S <i>server</i>]     [-I <i>interface</i>]     <i>sort_order</i>     [ <i>charset</i> ]</pre> <p>Or</p> <pre>charset -v</pre>
Parameters	<p><b>-P <i>password</i></b> specifies your password. If you do not specify <b>-P</b>, charset prompts for your password.</p> <p><b>-S <i>server</i></b> specifies the name of the server on which to change the character set and sort order.</p> <p><b>-I <i>interface</i></b> specifies the network interface used by the server.</p> <p><b><i>sort_order</i></b> specifies the name of the sort order file Adaptive Server will use.</p> <p><b><i>charset</i></b> specifies the character set Adaptive Server will use.</p> <p><b>-v</b> displays the version number and copyright message for charset.</p>
Usage	<ul style="list-style-type: none"> <li>Before using charset, you must set your SYBASE environment variable to point to the current release directory.</li> </ul>
Permissions	You must be a System Administrator to use charset.
See also	<p><b>Commands</b> set</p> <p><b>Utilities</b> langinstall</p>

## cobpre

Description	Precompiler for COBOL, located in <code>\$SYBASE/\$SYBASE_OCS/bin</code> ( <code>%SYBASE%\%SYBASE_OCS%\bin</code> in Windows). For a full description of <code>cpre</code> , see Appendix A of the <i>Open Client and Open Server Programmer's Supplement</i> .
Syntax	See above.

## **cpre**

Description	Precompiler for C, located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code> ( <code>%SYBASE%\%SYBASE_OCS%\bin</code> in Windows). For a full description of <code>cpre</code> , see Appendix A of the <i>Open Client and Open Server Programmer's Supplement</i> .
Syntax	See above.

## dataserver

**Description** **UNIX platforms only** The executable form of the Adaptive Server program, located in `$$SYBASE/$SYBASE_ASE/bin`.

**Syntax**

```
dataserver [-f] [-g] [-G] [-h] [-H] [-m] [-q] [-v] [-X]
           [-a path_to_CAPs_directive_file]
           [-b master_device_size [k | K | m | M | g | G | t | T ]
           [-c config_file_for_server]
           [-d device_name]
           [-e path_to_error_log]
           [-i interfaces_file_directory]
           [-K keytab_file]
           [-L config_file_name_for_connectivity]
           [-M shared_memory_repository_directory]
           [-N licinstan]
           [-n sa_login_name]
           [-p sa_login_name]
           [-r mirror_disk_name]
           [-s server_name]
           [-T trace_flag]
           [-u sa/sso_name]
           [-w master | model database]
           [-y [password] ]
           [-Z [size_spec] ]
           [-z page_size [ k | K ] ]
```

Or

```
dataserver -v
```

**Parameters**

- f  
forces initialization of a device or database. -f is valid only when used with -b and/or -w. The server fails to boot if you use -f without either -b or -w. -f forces the server in different ways, depending whether -w is present. See “Potential issues of using -f and -w options together” on page 161 and “Dependencies and conditions of -b and -w options” on page 161 for more information.
- g  
turns off event-logging.
- G *logserv\_name*  
specifies the name of the event log server.
- h  
prints this help message, then exists.

- H  
starts the High Availability (HA) server, if you have the HA feature installed on your Adaptive Server.
- m  
starts Adaptive Server in single-user mode.
- q  
treats quiesced databases as “in recovery.”
- v  
prints the version number and copyright message for dataserver, then exits.
- X  
starts this server as sybmon, not dataserver.
- a *path\_to\_CAPs\_directive\_file*  
specifies the path to the CAPs directive file.
- b *master\_device\_size* [ k | K | m | M | g | G | t | T ]  
specifies the size of the master device or database you want to build. The server calculates the sizes, so you can use “K”, “M”, “G”, and “T” instead of exact byte numbers.
- c *config\_file\_for\_server*  
specifies the full path name of an Adaptive Server configuration file. Use this parameter to start Adaptive Server with the configuration values in the specified configuration file.  
  
If you specify a configuration file with the `dataserver -c` parameter, make sure all the parameters in this configuration file are compatible before you boot the server. If some of the configuration parameters are incompatible, the server may not boot. To avoid this, do not specify a configuration file when you build the master device. The build phase uses all default settings when you do not specify a configuration file.  
  
For more information, see the *System Administration Guide*.
- d *device\_name*  
is the full path name of the device for the master database. The master database device must be writable by the user who starts Adaptive Server. If you do not use the `-d` parameter, the default master database device name is `d_master`.
- e *errorlogfile*  
is the full path name of the error log file for Adaptive Server system-level error messages.

- i *interfaces\_file\_directory*  
specifies the directory location of the interfaces file to search when connecting Adaptive Server. If -i is omitted, *dataserver* looks for a file named *interfaces* in the directory pointed to by your SYBASE environment variable.
- K *keytab\_file*  
specifies the path to the keytab file used for authentication in DCE.
- L *config\_file\_name\_for\_connectivity*  
specifies the name the configuration file for connectivity.
- M *sharedmem\_directory*  
places shared memory files in the specified directory instead of in the default location, *\$SYBASE*. If *sharedmem\_directory* starts with “/”, the directory name is assumed to be absolute. Otherwise, the directory name is interpreted relative to *\$SYBASE*.
- N *licinstant*  
specifies a nondefault directory location for the license cache file. The default location is *\$SYBASE/\$SYBASE\_ASE/sysam/server\_name.properties*.
- p *sso\_login\_name*  
specifies the login name of a System Security Officer when starting Adaptive Server, for the purposes of getting a new password for that account. Adaptive Server generates a random password, displays it, encrypts it, and saves it in *master..syslogins* as that account’s new password.
- r *mastermirror*  
starts the mirror of the master device. Use this parameter to start Adaptive Server if the master device has been damaged.
- s *servername*  
specifies the name of the Adaptive Server to start. If -s is omitted, a server named SYBASE is started.
- T *trace\_flag*
- u *sa/sso\_name*  
specifies the System Administrator or System Security Officer’s name you want to unlock.
- w [master | model]  
specifies whether you want to write a master or model database.



`-y [password]`

allows you to assign a password for the encrypted private key, so that the server prompts the user for a password. This password should match the password you used to encrypt the private key when it was created. You cannot use this parameter when you are running the server in the background.

---

**Note** Although you can set a password with `-y`, for security reasons Sybase strongly discourages you from doing so.

---

A private key is included with your server's digital certificate. By default, the certificate file located at  
`/usr/local/sybase/certificates/<servername>.cert`.

The location of the certificate file changes if you invoke the `sp_ssladmin addcert` command.

`-Z [size_spec]`

specifies the size of the initial master database.

`-z page_size [ k | K ]`

specifies the page size of the server. You must use `-b` and `-w` to use this flag, and name an even power of two between 2K and 16K, or else the server does not boot.

## Examples

**Example 1** Creates a new installation with a 100MB master device and a 4K page:

```
dataserver -d my_master_device -z 4k -b 100.02M
```

The spaces between options and their following arguments are optional and acceptable. This example specifies "100.02M" for a 100MB master device because the server requires 16K of overhead for its configuration area.

**Example 2** Rewrites a corrupt model database:

```
dataserver -d d_master -w model -s server_name
```

**Example 3** Rewrites a corrupt master database, specifying device size:

```
dataserver -d my_master_device -w master -z 4k
```

**Example 4** Rewrites a corrupt master database, specifying device and page sizes, forcing the server to accept these values in preference to what it may find in the config block:

```
dataserver -d my_master_device -w master -z 4k -b  
100.02M -f
```

**Example 5** Rewrites a corrupt master database, specifying a page size that does not match what the server finds in its config block. This produces a failure:

```
dataserver -d my_master_device -w master -z 8k
00:00000:00000:2001/01/19 12:01:26.94 server The
configured server page size does not match that
specified on the command line. To use the configured
size, omit the command line size; to use the command
line size, specify 'force' (-f).
```

**Example 6** Rewrites a corrupt master database, specifying an incorrect page size, even in a normal restart. This produces a failure:

```
dataserver -d my_master_device -z4000
dataserver: the 'z' flag may not be used without 'b' or
'w'. dataserver: server will ignore the 'z' flag.
dataserver: the 'z' flag contained an invalid page size.
dataserver: the page size must be an even power of two
between 2048 and 16384 bytes, inclusive.
```

## Usage

- **dataserver** allows you to create devices and databases that are up to 32Gb in size, depending on the limitation of your operating system. For more information on size limits, see the installation guide for your platform.
- Start Adaptive Server with the `startserver` command rather than by directly executing the `dataserver` program. If you need to change any of the default values, edit the `RUN_servername` file in your Sybase installation directory. See the `startserver` reference page for details.
- Because Adaptive Server passwords are encrypted, you cannot recover forgotten passwords. If all System Security Officers lose their passwords, the `-p` parameter generates a new password for a System Security Officer account. Start Adaptive Server with `-p`, immediately log in to Adaptive Server with the new random password, and execute `sp_password` to reset your password to a more secure one.
- When you use the `-w` parameter, `dataserver` uses the `ascii-8` character set instead of the `iso_1` character set. If you require the `iso_8` character set for master, you can load a dump of the master database or change the character set with `sqlloc` (`sqlloc` requires the `sybssystemprocs` database.)
- After you have finished running the Adaptive Server installation program, set the file permissions on the `dataserver` executable to limit who can execute it.

- If you do not specify an Adaptive Server name with the `-s` parameter, and you have not set the `DSLISTEN` environment variable, `dataserver` uses the default Adaptive Server name `SYBASE`. The value of the `DSLISTEN` environment variable overrides this default value, and the `-s` parameter overrides both the default and the `DSLISTEN` environment variable.
- Automatic login lockouts can cause a site to end up in a situation in which all accounts capable of unlocking logins (System Administrators and System Security Officers) are locked. If this occurs, use the `dataserver` utility with the `-u` parameter to check the specified login for System Administrator or System Security Officer authorization, unlock the account, and reset the value of the current failed logins counter to zero.

#### Potential issues of using `-f` and `-w` options together

Be particularly careful when using the `-f` and `-w` options together. When rewriting master database using the `-w` option, the server requires that the configuration block page size and device size are correct. If you do not provide them on the command line they must agree. The server refits the master device, and puts master and all other included databases back in their proper places.

When you use the `-f` option force initialization, your page size and master device size overrides those in the configuration block. In addition, `-f` assigns all other unknown space—allocation blocks that are either unused or are corrupted—to the master database.

#### Dependencies and conditions of `-b` and `-w` options

The effect of `-b` changes depending on whether `-w` is present:

- `-b` without `-w` creates a new master device as named by `-d` (the default is `d_master`) and with the page size as specified by `-z` (the default is 2048):
  - If the named device already exists as an OS file, the attempt fails, and you see a message such as:
 

```
File already exists. You must remove the existing
file before attempting to create a new one using
the server's -b option.
Unable to create master device.
```
  - If the named device names an existing raw partition, the attempt fails unless you include the `-f` flag. This reinitializes the raw partition as a server master device.
- `-b` with `-w master` tells `dataserver` to use the size specified in `-z` for the master device when recreating the master database. It implies nothing about creating a new device.

-w may or may not require additional flags if you use:

- -w model – the -z and -b flags are accepted but ignored.
- -w master for **new** installations – -z and -b are not required because the device size information is stored in the *config\_block*.
- -w master to **upgrade** older installations:
  - The server requires -b and/or -z if the *config\_block* does not contain a valid entry for the associated size(s). The command fails if it cannot get valid data for the page size or device size.
  - You can provide -b and/or -z when the *config\_block* contains valid entries for the size(s) they represent. However if the sizes do not match what is in the *config\_block*, you must add -f to force your new size preferences.

Permissions

Anyone with execute permission on the binary, and who has read/write access to all the files.

See also

**Commands** disk mirror, disk remirror, disk unmirror

**System procedures** sp\_ssladmin addcert

**Utilities** startserver

## ddlgen

### Description

A Java-based tool that generates definitions for server- and database-level objects in Adaptive Server. Adaptive Server version 11.9.2 and later supports ddlgen.

You must have either an `sa_role` or `sso_role` to generate DDL for:

- Encryption keys
- Logins
- Roles

For all other objects, users do not need any specific permissions or roles to generate DDL.

The command-line version of ddlgen is located in `$SYBASE/ASEP/bin` for UNIX, and `%SYBASE%\ASEP\bin` for Windows.

### Syntax

ddlgen

```
-Ulogin
-Ppassword
-S[[ssl:]server | host_name : port_number]
[-I interfaces_file]
[-T object_type]
[-Nobject_name]
[-Ddbname]
[-Xextended_object_type]
[-Ooutput_file]
[-Error_file]
[-Lprogress_log_file]
[-Jclient_charset]
-F[ % | SGM | GRP | USR | R | D | UDD | U | V |
  P | XP | I | RI | KC | TR | PC ]
```

Or

```
ddlgen -v
```

### Parameters

`-U login`

specifies a login name, and is case-sensitive.

`-P password`

specifies your password.

If you do not include the `-P` parameter in your ddlgen statement, ddlgen prompts you to specify a password.

For instructions on how to hide passwords, see “Hiding passwords in ddlgen” on page 180.

**-S** *[[ssl:] server | host\_name : port\_number]*  
specifies the name of the Adaptive Server. ddlgen looks this name up in the interfaces file or LDAP configuration file. If you specify:

- **[ssl:]** – allows you to generate DDL for objects in SSL-enabled servers. This parameter is optional.
- **-S [host\_name:port\_number]** – ddlgen uses the *host\_name* and *port\_number* provided, and neither interfaces nor LDAP configuration files are read or parsed.
- **-S[server] -l** – ddlgen parses the interfaces file specified at the user location for the server name (see the **-l** parameter description for more information).
- **-S[server]** – without specifying an interfaces file, ddlgen does the following:
  - a ddlgen first tries to read the LDAP configuration file from the standard location
  - b If the LDAP file does not exist, or exists but does not contain an Adaptive Server entry, then the interfaces file is parsed at its standard location for the server name
  - c If the LDAP file exists, then ddlgen uses it to search the server name. The interfaces file is not parsed, and the LDAP configuration file is parsed.

---

**Note** You must use the **-S** option because ddlgen does not connect to a default server.

---

**-l**  
specifies the interfaces file name, and corresponds to *\$\$SYBASE/interfaces* for UNIX, and *%SYBASE%\ini\sql.ini* for Windows. Use this optional parameter with **-S**.

**-T***object\_type*  
specifies the type of object you are creating. If you do not use **-T**, ddlgen generates DDL for the default database of login. The object types for **-T** are:

Object type	Description
C	cache
D	default
DB	database
DBD	database device

<b>Object type</b>	<b>Description</b>
DPD	dump device
EC	execution class
EG	engine group
EK	encrypted keys For more information on encrypted keys, see “Using ddngen for encrypted columns” on page 180
GRP	group
I	index
KC	key constraints
L	login
LK	logical key
P	stored procedure
R	rule
RI	referential integrity
RO	role
RS	remote server
SGM	segment
TR	trigger
U	table
UDD	user-defined datatype
USR	user
V	view
WS	user-defined Web service
WSC	Web service consumer
XP	extended stored procedure

**-Nobject\_name**

specifies the fully qualified name of the object you are creating, such as *-Ndb\_name.owner\_name.table\_name.object\_name*. The **-N** option:

- Is required if you specify any *object\_type* other than DB (database) in the **-T** parameter.
- Accepts wildcards with the use of %.
- Generates DDL for a trigger for a table, using the *-Ndb\_name.table\_owner.table\_name.trigger\_name* format.

To generate **all** triggers for a table, substitute *trigger\_name* with % using the *-Ndb\_name.table\_owner.table\_name.%* format.

- Generates DDL for all items of a specific object type on your server.
- Enforces strict order in which it parses the names in the *-Ndb\_name.owner\_name.table\_name.object\_name* format. If you only provide three arguments, ddlgen assumes they are *owner\_name*, *table\_name*, and *object\_name*, in that order. Alternatively, you can also use *-Nowner\_name.table\_name-Ddb\_name*. ddlgen does not impose this restriction if *object\_name* is an index (I).

**-Ddbname**

specifies the name of the database for the object you specify in the **-N** option. The default is the user's default database.

You cannot use the **-D** parameter when generating DDL for all triggers of a table.



**-Xextended\_object\_type**

differentiates the following:

- User tables (OU) from proxy tables (OD) when you specify a table as your object type (-TU)
- Temporary databases (OD) from normal databases (OU) or archive databases (OA) when you specify database as your object type (-TDB)
- SQLJ procedures (OD) from stored procedures (OU) when you specify procedure as your object type (-TP).

If *object\_type* (-T) is U (table) and -X is not specified, ddlgen generates DDL for both user tables and proxy tables. To generate DDL only for:

- **user tables** – use the OU extended object type with the -X option.
- **proxy tables** – use the OD extended object type with the -X option.

---

**Note** ddlgen does not support schema generation for system tables.

---

**-Ooutput\_file**

specifies an output file for the generated DDL. If you do not specify -O, the DDL you create appears in a console window.

**-Error\_file**

specifies a log file for recording errors. If you do not specify -E, the generated errors appear in a console window.

**-Lprogress\_log\_file**

specifies a log file for recording the progress of ddlgen. If you do not specify -L, the progress is not recorded.

**-Jclient\_charset**

specifies the character set to use on the client. -J*client\_charset* requests that Adaptive Server convert to and from *client\_charset*, the character set used on the client. A filter converts input between *client\_charset* and the Adaptive Server character set.

Omitting -J sets the character set to a default for the platform. The default may not necessarily be the character set that the client is using.

---

**Note** For HP platforms – you **must** use -Jiso\_1 to specify the correct character set.

---

-F

filters out indexes, triggers, and constraints out of table and database definitions in the DDL of table- and database-level objects. The valid filters are:

- **For tables** – [ % | I | RI | KC | TR | PC ]
- **For databases** – [ % | SGM | GRP | USR | R | D | UDD | U | V | P | XP | I | RI | KC | TR ]

The filter options are:

Option	Filters out
%	Everything, and retrieves the schema-only definition of a database or table.
SGM	Segments
GRP	Groups
USR	Users
R	Rules
D	Defaults
UDD	User-defined datatypes
U	User tables
V	Views
P	Stored procedures
PC	Partition condition
XP	Extended stored procedures
I	Indexes
RI	Referential integrity constraints
KC	Primary- and unique-key constraints
TR	Triggers

If you use an invalid filter parameter, `ddlgen` generates a warning, ignores that parameter, and continues with the rest of the valid parameters you specify.

If you specify % along with other filter parameters, `ddlgen` ignores all other filterable parameters, and only shows schema-only definitions. `ddlgen` then continues to evaluate the dependencies within the subset of the applied as the filterable parameters for the database.

-v

displays the version and copyright message of `ddlgen` and returns to the operating system.

## Examples

**Example 1 Caches** – Generates DDL for a cache called *default data cache* on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TC -N"default data cache"
```

To generate DDL for all caches:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TC -N%
```

**Example 2 Defaults** – Generates DDL for a default called “phondflt” owned by jones in the pubs2 database on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TD -Njones.phondflt -Dpubs2
```

Alternatively, because `ddlgen` allows you to use a fully qualified name in the `-N` flag, you can omit the `-Ddbname` and include the database name in the `-N` option:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TD -Ndbname.owner.defaultname
```

To generate DDL for all defaults in a database owned by “owner”:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TD -Nowner.% -Ddbname
```

**Example 3 Databases** – Generates DDL for a database called *pubs2* on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TDB -Npubs2
```

If you do not specify a *dbname*, `ddlgen` generates DDL for the default database of *login*:

```
ddlgen -Ulogin -Ppassword -Sserver:port
```

If you do not use the `-T` parameter, `ddlgen` generates DDL for a default-type database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -Ndbname
```

To generate DDL for all databases:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -N%
```

**Example 4 Database device** – Generates DDL for a database device called *master* running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TDBD -Nmaster
```

To generate DDL for all database devices:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDBD -N%
```

**Example 5 Temporary databases** – Generates DDL for all databases, including *tempdb*:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -N%
```

To generate DDL for all temporary databases, use the OD extended database type:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -XOD -N%
```

Although you can use the OD extended type in Adaptive Server versions 12.5.0.3 and later, versions earlier than 12.5.0.3 issue warning messages. You can safely ignore this message; ddlgen continues processing the command.

To generate DDL for all databases except temporary databases, use the OU extended type:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -XOU -N%
```

The following generates DDL for a temporary database named tempdb1:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -Ntempdb1
```

The output includes the following:

- A create temporary database statement

```
create temporary database tempdb1 on master = 4,  
asdas = 2  
go
```

- An sp\_tempdb bind statement where the isql application is bound to tempdb1:

```
sp_tempdb 'bind','ap', 'isql', 'DB', 'tempdb1'  
go
```

---

**Note** DDL for objects such as views, stored procedures, and tables is not generated along with DDL for a temporary database because these objects are temporary, and are re-created when the server restarts.

When you use the -F parameter to filter a table while generating DDL for a database object, then indexes, referential integrity, key constraints and triggers automatically get filtered, as they are a subset of the table object.

---

**Example 6 Dump device** – generates DDL for a dump device called tapedump1 running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TDPD -Ntapedump1
```

To generate DDL for all dump devices:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDPD -N%
```

**Example 7 Encrypted keys** – generates system encryption passwords along with DDLs for encryption keys when you include the extended option `-XOD`. The output generates the `sp_encryption` statement followed by DDL statements for all encrypted keys. This example generates DDL for the login “george” on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEK -Ngeorge -XOD
```

To generate DDL for all the encrypted keys associated with in the database authors database on a machine named HARBOR using port 1955:

The fully qualified name for an encrypted key is `dbName.owner.keyName`. Using the `EK` option generates both DDL and encrypted column information of a table.

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEK -Nauthors.dbo.%
```

**Example 8 Engine groups** – generates DDL for an engine group called LASTONLINE running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEG -NLASTONLINE
```

To generate DDL for all engine groups:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TEG -N%
```

**Example 9 Execution class** – generates DDL for an execution class called EC2 running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEC -NEC2
```

To generate DDL for all execution classes:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TEC -N%
```

**Example 10 Extended stored procedures** – generates DDL for the `xp_cmdshell` extended stored procedure in the `pubs2` database, owned by Jones and running on a machine named HARBOR using port 1955, by using the fully qualified `dbName.owner.extendedstoredprocedure` format with the `-N` option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TXP -Npubs2.jones.xp_cmdshell
```

Alternatively, you can use the `-D` option instead of using the fully qualified name:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TXP
      -Nowner.extendedstoredprocedure -Ddbname
```

To generate DDL for all extended stored procedures:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TXP -Ndbname.owner.%
```

**Example 11 Filters** – Generates DDL for the authors table in the pubs2 database, filtering for all indexes (I), and referential integrity constraints (RI), primary and unique key constraints (KC), triggers (TR), and partition condition (PC) from the DDL of a table:

```
ddlgen -Uroy -Proyl23 -TU -Nauthors -Dpubs2 -F%
```

Alternatively, you can specify each of the filters individually:

```
ddlgen -Ulogin -Ppassword -TU -Ndbname.owner.table  
-FI,RI,KC,TR
```

The following generates the definition of *table\_name* while filtering out foreign keys and primary-unique keys:

```
ddlgen -Ulogin -Ppassword -TU -Ntable_name -Ddbname  
-FRI,KC
```

Both of these generate foreign keys for a specified user in the entire database:

```
ddlgen -Ulogin -Ppassword -TRI -N%.%.% -Ddbname
```

Or:

```
ddlgen -Ulogin -Ppassword -TRI -Ndbname%.%.%
```

Both of these generate DDL for the primary and unique keys of all the tables in a database that begin with “PK”:

```
ddlgen -Ulogin -Ppassword -TKC -Ndbname.%.%.PK%
```

Or:

```
ddlgen -Ulogin -Ppassword -TKC -N%.%.PK% -Ddbname
```

The following generates schema-only definition of a database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TF -Ndbname  
-F%
```

Alternatively, you can specify each of the filters individually:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -Ndbname  
-FSGM,GRP,USR,R,D,UDD,V,P,XP,I,RI,KC,TR
```

The following generates the database DDL skipping the compiled object:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -Ndbname  
-FTR,D,XP,V,R
```

The following generates database definition without a table definition:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -Ndbname  
-FU
```

**Example 12 Groups** – Generates DDL for a group called “public” in the pubs2 database, running on a machine named HARBOR using port 1955, by using the fully qualified *dbname.groupname* format in the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TGRP -Npubs2.public
```

Alternatively, you can use the -D option to specify the *dbname*:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TGRP -Ngroupname -Ddbname
```

To generate DDL for all groups:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TGRP -Ndbname.%
```

**Example 13 Indexes** – Generates DDL for an index called au\_lname for the table authors owned by dbo, in the pubs2 database:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TI -Ndbo.authors.au_lname -Dpubs2
```

Alternatively, because *ddlgen* allows you to use a fully qualified name in the -N flag, you can omit the -D*dbname* and include the database name in the -N option:

```
ddlgen -Ulogin -Ppassword -Sserver:port
-TI -Ndbname.owner.tablename.indexname
```

If you use a fully qualified name, you may omit the -D option.

To generate DDL for all indexes for a single table:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TI
-Ndbname.owner.tablename.%
```

To generate DDL for all indexes of all tables in a database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TI
-Ndbname.%.%.%
```

For example, this generates DDL for all indexes for all tables in the pubs2 database:

```
ddlgen -Usa -P -SHARBOR:1955 -TI -Npubs2.%.%.%
```

**Example 14 Keys** – Both of these generate DDL for the primary and unique keys of all the tables in a database that begin with “PK”:

```
ddlgen -Usa -P -TKC -Ndbname.%.%.PK%
```

Or:

```
ddlgen -Usa -P -TKC -N%.%.PK% -Ddbname
```

**Example 15 Logical keys** – LK generates logical keys of table defined by `sp_primarykey`, `sp_commonkey`, `sp_foreignkey` statements. Since these keys do not have a name, the name of the object in this case would be the name of the table. This example generate a DDL for logical keys of table authors in database pubs2 running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TLK -Npubs2.dbo.authors
```

To generate DDL for all logical keys in database pub2 use:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TLK -Npubs2.%.%
```

To filter out logical keys definition from DDL of table authors use LK in -F argument, use:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TLK -Npubs2.dbo.authors -FLK
```

**Example 16 Logins** – TL generates DDL for one or all logins. This example generates DDL for all logins on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TL -N%
```

---

**Note** The password in the DDL generated for all logins is “password”.

---

Alternatively, you can specify an individual login by using `-Nusername` instead of `-N%`:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TL -Nusername
```

If server-wide password complexity options have been specified for the login or logins, all `sp_addlogin` and `sp_loglogin` DDL statements are generated first, followed by DDL statements for the password options. This example generates DDL for the login “george” on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TL -Ngeorge
```

**Example 17 Remote Servers** – Generates DDL for a remote server called ORANGE on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TRS -NORANGE
```

To generate DDL for all remote servers:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TRS -N%
```

**Example 18 Roles** – Generates DDL for the sa\_role on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TRO -Nsa_role
```

To generate DDL for all roles:



```
ddlgen -Ulogin -Ppassword -Sserver:port -TRO -N%
```

---

**Note** The password in the DDL generated for all roles is “password”.

---

**Example 19 Rules** – Generates DDL for all rules associated with authors on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TR -Nauthors.dbo.%
```

The % symbol tells `ddlgen` to create DDLs for all rules that exist on the server.

You can also give the fully qualified name of the rule:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TR -Ndbname.owner.rulename
```

Alternatively, you can also use the `-D` parameter:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TR -Nowner.rulename -Ddbname
```

**Example 20 Segments** – Generates DDL using the fully qualified `dbname.segmentname` format in the `-N` option for a segment called `logsegment` for the `pubs2` database, on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TSGM -Npubs2.logsegment
```

Alternatively, you can use specify the `dbname` using the `-D` option:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TSGM -Nsegmentname -Ddbname
```

To generate DDL for all segments:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TSGM -Ndbname.%
```

**Example 21 SQLJ functions** – Generates DDL for a SQLJ function named `region_of` owned by `dbo` in database master:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TF -Nmaster.dbo.region_of
```

Alternatively you can also use the `-D` parameter:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TF -Ndbo.region_of -Dmaster
```

To generate DDL for all SQLJ functions in a database, use object type `F`:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TF -Ndbname.owner.%
```

**Example 22 SQLJ procedures** – Are a kind of stored procedure. You generate DDL for SQL procedures along with DDL for stored procedures. The following generates DDL for all stored procedures—including SQLJ procedures—owned by `dbo` in the master database:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TP -Nmaster.dbo.%
```

To generate DDL for all SQLJ procedures that are only owned by dbo in the master database, use the following, where the extended type OD refers to SQLJ procedures:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TP -Nmaster.dbo.% -XOD
```

To generate DDL for all procedures except SQLJ procedures owned by dbo in the master database, use the following, where the extended type OU refers to all stored procedures except SQLJ procedures:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TP -Nmaster.dbo.% -XOU
```

**Example 23 Stored procedures** – Generates DDL for the sp\_monitor stored procedure for the pubs2 database on a machine named HARBOR using port 1955, using the fully qualified *dbname.owner.procedure\_name* format for the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TP -Npubs2.dbo.sp_monitor
```

Alternatively, you can use specify the *dbname* using the -D option:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TP -Nowner.procedurename -Ddbname
```

To generate DDL for all stored procedures:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TP -Ndbname.owner.%
```

**Example 24 SSL-enabled servers** – Generates DDL for objects in the pubs2 database for an SSL-enabled Adaptive Server running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -Sssl:HARBOR:1955 -TDB -Npubs2
```

**Example 25 Tables** – Generates DDL for all user tables in the pubs2 database owned by “dbo” and running on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TU -Ndbo.% -Dpubs2
```

You can also use the -N parameter to give the fully qualified name of the table:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TU  
-Ndbname.tableowner.tablename
```

Alternatively, you can also use the -D parameter to specify the database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TU  
-Ntableowner.tablename -Ddbname
```

To generate DDL for all proxy tables, which uses the value OD, use -XOD instead, where X is the extended type, and OD denotes proxy tables:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TU  
-Ntableowner.% -Ddbname -XOD
```

To generate DDL for all user tables, which uses the value OU, use -XOU instead, where X is the extended type, and OU denotes user tables:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TU
      -Ntableowner.% -Ddbname -XOU
```

To generate DDL for all tables, including user tables and proxy tables:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TU -Ndbname.tableowner.%
```

**Example 26 Triggers** – Generates DDL for the trigger checksum for the pubs2 database on a machine named HARBOR using port 1955, using the fully qualified *dbname.owner.trigger\_name* format for the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TTR -Npubs2.dbo.checksum
```

Alternatively, you can use specify the *database\_name* using the -D option:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TTR
      -Nowner.triggername -Ddbname
```

You can also generate DDL for a trigger for a table, using:

```
-Ndb_name.table_owner.table_name.trigger_name
```

To generate DDL for all triggers of a database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TTR -Ndbname.owner.%
```

You can also use the following format to generate DDL for all triggers of a table:

```
-Ndb_name.table_owner.table_name.%
```

---

**Note** You cannot use the -D parameter when generating DDL for all triggers of a table.

---

**Example 27 User-defined datatypes** – Generates DDL for the user-defined datatype “Identype” for the pubs2 database on a machine named HARBOR using port 1955 using the fully qualified *dbname.userdefined\_datatype* format for the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TUDD -Npubs2.Identype
```

Alternatively, you can use the -D option to specify the *dbname*:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TUDD
      -Nuserdefined_datatype -Ddbname
```

To generate DDL for all user-defined datatypes:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TUDD -Ndbname.%
```

**Example 28 Views** – Generates DDL for a view named retail owned by Miller in the pubs2 database running on a machine named HARBOR using port 1955, by using the fully qualified *dbname.owner.viewname* format with the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TV -Npubs2.miller.retail
```

Alternatively, you can use the -D option instead of using the fully qualified name:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TV -Nowner.viewname -Ddbname
```

To generate DDL for all views:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TV -Ndbname.owner.%
```

**Example 29 Users** – Generates DDL for a user named Smith in the pubs2 database running on a machine named HARBOR using port 1955, by using a fully qualified *dbname.username* format with the -N option:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TUSR -Npubs2.smith
```

Alternatively, you can use both the -N and -D options instead of using a fully qualified name in -N:

```
ddlgen -Ulogin -Ppassword -Sshost_name:port -TUSR -Nusername -Ddbname
```

To generate DDL for all users:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TUSR -Ndbname.%
```

**Example 30 User-defined Web services** – Generates DDL for a named user-defined Web service, *sp\_who\_service*, in the pubs2 database running on a machine named HARBOR using port 1995, by using a fully qualified *dbname.username.webservice\_name* format with the -N and -T options:

```
ddlgen -Uroy -Proy123 -SHARBOR:1995 -TWS  
-Npubs2.dbo.sp_who_service
```

The syntax for generating DDL for a named user-defined Web service is as follows:

```
ddlgen -Ulogin -Ppassword -Sshost_name:port -TWS  
-Ndbname.owner.webservice_name
```

To generate DDL for all user-defined Web services owned by all users in database *dbname*:

```
ddlgen -Ulogin -Ppassword -Sshost_name:port -TWS -Ndbname.%.%
```

---

**Note** An *sp\_webservices* 'addalias' statement is only generated if the DDL is to be generated for all user-defined web services or for a database.

---

- Usage
- `ddlgen` does not identify existing sequences within views, stored procedures or triggers. For this reason, when generating DDL for a database, you must first run `ddlgen` on those views, stored procedures and triggers that are independent, before running `ddlgen` on those with dependencies. For example, if view B depends on view A, you must first run `ddlgen` on view A, before running it on view B.
  - The default information for `ddlgen` is:

Option	Parameter	Required	Default
-U	<i>username</i>	Yes	None
-P	<i>password</i>	Yes	None
-S	<i>host_name:port_number</i>	Yes	None
-T	<i>object_type</i> See the -T parameter description for a list of valid object types	No	Database
-N	<i>object_name</i>	Yes, if <i>object_type</i> for -T is not DB (database)	Default database name of <i>username</i> , if -T <i>object_type</i> is db or if -T is not specified
-D	<i>database_name</i>	No	Default database of <i>username</i>
-X	<i>extended_object_type</i> Options are: <ul style="list-style-type: none"> <li>• OU – for user tables, user databases (excluding temporary databases), and stored procedures (excluding SQLJ procedures).</li> <li>• OD – for proxy tables, temporary databases, and SQLJ procedures.</li> </ul>	No; use only when the <i>object_type</i> for -T is: <ul style="list-style-type: none"> <li>• U (user table)</li> <li>• P (procedure)</li> <li>• DB (database)</li> </ul>	None
-O	<i>output_file_name</i>	No	Standard out
-E	<i>error_file_name</i>	No	Standard out
-L	<i>log_file_name</i>	No	None
-V	<i>version_number</i> of <code>ddlgen</code>	No	None

- At the command line, invoke `ddlgen` using the `ddlgen` shell script file (`ddlgen.bat` for Windows), included in your Adaptive Server installation. The main class in `DDLGen.jar` is `com.sybase.ddlgen.DDLGenerator`.
- To start `ddlgen` in the Sybase Central plug-in for Adaptive Server:
  - a Right-click on the object for which you want to generate DDL.
  - b Select Generate DDL.

- In the output DDL of create table, bind statements are generated as independent DDL instead of dependent DDL.

The following section provides more details about:

- “Hiding passwords in ddlgen” on page 180
- “Using ddlgen for encrypted columns” on page 180

#### Hiding passwords in *ddlgen*

When you issue the *ddlgen* utility in a Unix command-line environment, other users on that Unix machine can see your *ddlgen* command—including its password—if they issue the *ps* process management command, which shows the status of processes that are running on that machine.

Adaptive Server 15.0.2 ESD #2 includes a new *ddlgen* *-P* password parameter option that lets you to invoke *ddlgen* from a script so that the password is hidden from other users.

To achieve this, include the string “pwd” in the *-P* parameter. This acts as a pseudo password, while you supply the actual password in the next line of the script:

```
ddlgen -Ulogin -Ppwd -Sserver:port -Ttype -Nname << END
$PSWD
END
```

The actual *ddlgen* password string is hidden in the *\$PSWD* environment variable.

If you prefer to keep your password in a file, replace *\$PSWD* with '*cat filename*', where *filename* is the location of your password file. For example:

```
ddlgen -Ulogin -Ppwd -Sserver:port -Ttype -Nname << END
'cat filename'
END
```

#### Using *ddlgen* for encrypted columns

The *ddlgen* utility supports both pre-15.0.2 and 15.0.2 encryption.

Pre-15.0.2 *ddlgen*  
support

Pre-15.0.2 *ddlgen* support includes generating DDL for an encryption key in a database, and generating DDL to synchronize encryption keys across servers.

#### v **Generating DDL for an encryption key in a database**

- To generate DDL for an encryption key “*ssn\_key*” in a database called “*SampleKeysDB*,” use the following syntax:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key
```

Where:

- EK – is the encrypted key type
- SampleKeysDB.dbo.ssn\_key – is the fully qualified name for the encrypted key

ddlgen generates the following using this syntax:

```
-----
-- DDL for EncryptedKey 'ssn_key'
-----
print 'ssn_key'

create encryption key SampleKeysDB.dbo.ssn_key for AES
with keylength 128
init_vector random
go
```

#### v **Generating DDL to synchronize encryption keys across servers**

- To synchronize encryption keys across servers for data movement, use the ddlgen -XOD extended option to generate the create encryption key command that specifies the key's encrypted value as represented in sysencryptkeys, such as in this sample:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key -XOD
```

The output from this ddlgen command would be:

```
-----
---
-- DDL for EncryptedKey 'ssn_key'
-----
---
print 'ssn_key'

create encryption key SampleKeysDB.dbo.ssn_key for AES
with keylength 128
passwd 0x00009EECD959E00095A401
init_vector random
keyvalue
0x11D6B8AA6ACB0C0490A363791531DD6E6728C800FDD1A13BFF795FA22726C16101
keystatus 32
go
```

Adaptive Server  
15.0.2 *ddlgen* support

The ddlgen utility also supports 15.0.2 encryption keys.

**v Generating DDL without specifying the -XOD flag**

- For 15.0.2 encryption, two things can happen if users do not specify the -XOD flag in ddlgen:
  - If users **did not** specify a password when the encryption key was created, ddlgen generates DDL with no password.
  - If users specified a password when the encryption key was first created, ddlgen generates the default password of 'password'. This is similar to what ddlgen does for roles and login passwords, and its output looks similar to the following:

```
-----  
----  
-- DDL for EncryptedKey 'ssn_key'  
-----  
----  
print 'ssn_key'  
  
--The DDL is generated with a default password - 'password' as  
--a password was specified when this key was created.  
  
create encryption key SampleKeysDB.dbo.ssn_key for AES  
with keylength 128  
passwd 'password'  
init_vector random  
go
```

**v Generating DDL with the -XOD flag**

- In 15.0.2, when a user specifies “-XOD” in the ddlgen command, ddlgen generates DDL that includes a system encryption password (if it has been set and DDL is generated for a key encrypted with a system encryption password) and DDL for keys.

The following syntax generates a system encryption password:

```
ddlgen -Usa -P -Sserver -TEK -NsampleKeysdb.dbo.ek1 -XOD
```

The output for the command is:

```
-- System Encryption Password  
  
use SampleKeysDB  
go  
  
sp_encryption 'system_encr_passwd',  
'0x8e050e3bb607225c60c7cb9f59124e99866ca22e677b2cdc9a4d09775850f4721',  
NULL, 2, 0
```



```

go

-----
--
-- DDL for EncryptedKey 'ek1'
-----
--

print '<<<<< CREATING EncryptedKey - "ek1" >>>>>'
go

create encryption key SampleKeysDB.dbo.ek1 for AES
with keylength 128
passwd 0x0000C7BC28C3020AC21401
init_vector NULL
keyvalue
0xCE74DB1E028FF15D908CD066D380AB4AD3AA88284D6F7742DFFCADCAABE4100D01
keystatus 32
go

```

---

**Note** When migrating keys from a source to a target server using `ddlgen`, set the system encryption password to NULL (if it exists) in the target server if you want to run the `ddlgen` output (from the source server) for encryption keys generated using “-XOD” parameter. Failure to do this results in errors when you try to execute the `ddlgen` output against the target server.

---

**Key copy support** The `ddlgen` utility also generates DDL for key copies along with the DDL for base key. For example, the following syntax would generate DDL for “`ssn_key`” and its key copies:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key
```

The output from `ddlgen` would look like:

```

-----
-- DDL for EncryptedKey 'ssn_key'
-----

print 'ssn_key'

--The DDL is generated with a default password - 'password' as
--a password was specified when this key was created.

create encryption key SampleKeysDB.dbo.ssn_key for AES
with keylength 128
passwd 'password'

```

```
init_vector random
go

print 'Key Copies for ssn_key'

-- Generating DDL for Key Copies for 'ssn_key'

alter encryption key 'ssn_key'
with passwd 'password'
add encryption with passwd 'passwd'
for user 'dbo'.
```

If you include the -XOD flag, the DDL for key copy would look like:

```
alter encryption key SampleKeysDB.dbo.ssn_key add encryption
with keyvalue
0x84A7360AA0B28801D6D4CBF2F8219F634EE641E1082F221A2C58C9BBEC9F49B501
passwd 0x000062DF4B8DA5709E5E01
keystatus 257
for user 'user1'
go
```

EKC encryption key  
copy filter

With Adaptive Server 15.0.2, ddlgen supports the EKC (encryption key copy) extended type for its -F filter argument, to suppress the generation of key copies for encryption keys.

This example uses -FEKC to avoid creating DDL for key copies when generating DDL for the “ssn\_key” encryption key:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key -FEKC
```

The output from ddlgen would look like:

```
-----
---
-- DDL for EncryptedKey 'ssn_key'
-----
---
print 'ssn_key'

--The DDL is generated with a default password - 'password' as
--a password was specified when this key was created.

create encryption key SampleKeysDB.dbo.ssn_key for AES
with keylength 128
passwd 'password'
init_vector random
go
```

**Create table DDL**            `ddlgen` can generate `decrypt_default` statements (if set for an encrypted column) along with DDL of a table.

The following example issues a `ddlgen` command on a table called `employee` which has an “`ssn`” column that is encrypted with encryption key “`ssn_key`,” and a `decrypt default` value that is set to “100”:

```
ddlgen -Usa -P -Sserver -TU -Nemployee
```

The DDL output from the command is:

```
create table employee (
  ssn          int          not null  encrypt with ssn_key decrypt_default
100 ,
  last_name    int          not null ,
  first_name   int          not null
)
lock allpages
on 'default'
go
```

**Permissions**            Users must have either `sa_role` or `sso_role` to generate DDL for:

- Encryption keys
- Logins
- Roles

For all other objects, users do not need any specific permissions or roles to generate DDL.

## defncopy

**Description** Copies definitions for specified views, rules, defaults, triggers, or procedures from a database to an operating-system file or from an operating-system file to a database. Located in `$SYBASE/$SYBASE_OCS/bin`.

**Windows** The utility is `defncopy.exe` and is located in `%SYBASE%\%SYBASE_OCS%\bin`.

**Syntax**

```
defncopy [-X]
          [-a display_charsef]
          [-l interfaces_file]
          [-J [client_charsef]]
          [-K keytab_file]
          [-P password]
          [-R remote_server_principal]
          [-S [server_name]]
          [-U username]
          [-V security_options]
          [-Z security_mechanism]
          [-z language]
          { in file_name database_name |
            out file_name database_name [owner.]object_name
            [[owner.]object_name...] }
```

Or

```
defncopy -v
```

**Parameters**

-X

initiates the login with client-side password encryption in this connection to the server. `defncopy` (the client) specifies to the server that password encryption is desired. The server sends back an encryption key, which `defncopy` uses to encrypt your password, and the server uses to authenticate your password when it arrives.

If `defncopy` crashes, the system creates a core file which contains your password. If you did not use the encryption option, the password appears in plain text in the file. If you used the encryption option, your password is not readable.

**-a *display\_charset***

runs defncopy from a terminal whose character set differs from that of the machine on which defncopy is running. Use -a in conjunction with -J to specify the character set translation file (*.xlt* file) required for the conversion. Use -a without -J only if the client character set is the same as the default character set.

---

**Note** The *ascii\_7* character set is compatible with all character sets. If either the Adaptive Server character set or the client character set is set to *ascii\_7*, any 7-bit ASCII character can pass unaltered between client and server. Other characters produce conversion errors. See the *System Administration Guide* for more information on character set conversion.

---

**-l *interfaces\_file***

specifies the name and location of the interfaces file to search when connecting to Adaptive Server. If you do not specify -l, defncopy looks for a file named *interfaces* in the directory specified by the SYBASE environment variable in UNIX platforms, and *sql.ini* in the *ini* subdirectory for your Sybase release directory in Windows.

**-J *client\_charset***

specifies the character set to use on the client. A filter converts input between *client\_charset* and the Adaptive Server character set.

-J *client\_charset* requests that Adaptive Server convert to and from *client\_charset*, the client's character set.

-J with no argument sets character set conversion to NULL. No conversion takes place. Use this if the client and server are using the same character set.

Omitting -J sets the character set to a default for the platform. The default may not be the character set that the client is using. For more information about character sets and their associated flags, see the *System Administration Guide* and *Configuration Guide* for your platform.

**-K *keytab\_file***

specifies the path to the keytab file used for authentication in DCE.

**-P *password***

specifies your password. If you do not specify -P, defncopy prompts for your password.

- R *remote\_server\_principal***  
specifies the principal name for the server. By default, a server's principal name matches the server's network name (which is specified with the **-S** parameter or the **DSQUERY** environment variable). Use the **-R** parameter when the server's principal name and network name are not the same.
- S *server\_name***  
specifies the name of the Adaptive Server to which to connect. If you specify **-S** with no argument, defncopy looks for a server named SYBASE. If you do not specify **-S**, defncopy uses the server specified by your **DSQUERY** environment variable.
- U *username***  
specifies a login name. Login names are case sensitive. If you do not specify *username*, defncopy uses the current user's operating system login name.
- V *security\_options***  
specifies network-based user authentication. With this option, the user must log in to the network's security system before running the utility. In this case, users must supply their network user name with the **-U** option; any password supplied with the **-P** option is ignored.
- V** can be followed by a *security\_options* string of key-letter options to enable additional security services. These key letters are:
- **c** – Enable data confidentiality service
  - **i** – Enable data integrity service
  - **m** – Enable mutual authentication for connection establishment
  - **o** – Enable data origin stamping service
  - **r** – Enable data replay detection
  - **q** – Enable out-of-sequence detection
- Z *security\_mechanism***  
specifies the name of a security mechanism to use on the connection.
- Security mechanism names are defined in the *\$\$SYBASE/install/libtcl.cfg* configuration file. If no *security\_mechanism* name is supplied, the default mechanism is used. For more information on security mechanism names, see the description of the *libtcl.cfg* file in the *Open Client and Open Server Configuration Guide*.

**-z *language***

is the official name of an alternate language that the server uses to display defncopy prompts and messages. Without the -z flag, defncopy uses the server's default language.

Add languages to an Adaptive Server at installation, or afterwards with the utility langinstall (langinst in Windows) or the stored procedure sp\_addlanguage.

**in | out**

specifies the direction of definition copy.

***file\_name***

specifies the name of the operating system file destination or source for the definition copy. The copy out overwrites any existing file.

***database\_name***

specifies the name of the database to copy the definitions from or to.

***owner***

is optional if you or the Database Owner own the table being copied. If you do not specify an owner, defncopy first looks for a table of that name that you own, and then looks for one owned by the Database Owner. If another user owns the table, you must specify the owner name or the command fails.

***object\_name***

specifies name(s) of database object(s) for defncopy to copy out. Do not use *objectname* when copying definitions in.

**-v**

displays the version and copyright message of defncopy and returns to the operating system.

**Examples**

**Example 1** Copies definitions from the file *new\_proc* into the database stagedb on server MERCURY. The connection with MERCURY is established with a user of name "sa" and a NULL password:

```
defncopy -Usa -P -SMERCURY in new_proc stagedb
```

**Example 2** Copies definitions for objects sp\_calccomp and sp\_vacation from the employees database on the SYBASE server to the file *dc.out*. Messages and prompts display in french. The user is prompted for a password:

```
defncopy -S -z french out dc.out employees sp_calccomp sp_vacation
```

**Usage**

- Use this syntax for defncopy\_r if you are using threaded drivers.
- Use this syntax for defncopy you are using threaded drivers in the IBM platform.

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use defncopy.
- Invoke the defncopy program directly from the operating system. defncopy provides a noninteractive way to copy out definitions (create statements) for views, rules, defaults, triggers, or procedures from a database to an operating system file. Alternatively, it copies in all the definitions from a specified file.
- The in *filename* or out *filename* and the database name are required and must be stated unambiguously. For copying out, use file names that reflect both the object's name and its owner.
- defncopy ends each definition that it copies out with the comment:

```
/* ### DEFNCOPY: END OF DEFINITION */
```

Definitions created as text must end with this comment so that defncopy can copy them in successfully.

- Enclose values specified to defncopy in quotation marks, if they contain characters that could be significant to the shell.

---

**Warning!** Long comments of more than 100 characters that are placed before a create statement may cause defncopy to fail.

---

#### Permissions

- You must have select permission on the sysobjects and syscomments tables to copy out definitions; you do not need permission on the object itself.



- You may not have select permission on the text column of the syscomments table if the System Security Officer has reset the allow select on syscomments.text column parameter with the system procedure sp\_configure. This reset restricts select permission to the object owner and the System Administrator. This restriction is required in order to run Adaptive Server in the **evaluated configuration**, as described in the installation and configuration documentation for your platform. In this case, the object owner or a System Administrator must execute defncopy to copy out definitions.

---

**Note** If the text has been encrypted, it may be hidden from you even if you have all the required permissions. See “Verifying and Encrypting Source Text” in the *Transact-SQL User’s Guide* for more information.

---

- You must have the appropriate create permission for the type of object you are copying in. Objects copied in belong to the copier. A System Administrator copying in definitions on behalf of a user must log in as that user to give the user proper access to the reconstructed database objects.

Tables used

syscomments, sysobjects

See also

**Commands** create, select

**System procedures** sp\_addlanguage, sp\_checkreswords, sp\_configure, sp\_procqmode, sp\_remap

**Utilities** langinstall

## dscp

Description	<b>UNIX platforms only</b> Allows you to view and edit server entries in the interfaces file from the command line in UNIX platforms. Located in <code>\$\$SYBASE/\$\$SYBASE_OCS/bin</code> .
Syntax	<code>dscp [-p]</code> or <code>dscp -v</code> To exit from dscp: <code>quit</code> or <code>exit</code>
Parameters	<code>-p</code> suppresses command-line prompts. <code>-v</code> displays the version and copyright message of dscp and returns to the operating system.
Examples	Opens the default interfaces file for editing and suppresses the command-line prompt: <code>dscp -p</code>
Usage	<ul style="list-style-type: none"><li>• You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use dscp.</li><li>• The dscp utility program is a text-based utility.</li><li>• See Chapter 6, “Using dscp” for more information about the dscp utility program.</li></ul>
See also	<b>Utilities</b> dsedit

## dsedit

Description	<p><b>UNIX platforms</b> The dsedit utility allows you to view and edit server entries in the interfaces file using a GUI based on X11/Motif in UNIX platforms. The utility is located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code>.</p> <p><b>Windows</b> The dsedit.exe utility creates and modifies network connection information in the interfaces file. The utility is located in <code>%%SYBASE%\%%SYBASE_OCS%\bin</code>.</p>
Syntax	<pre>dsedit or dsedit -v</pre>
Parameters	<pre>-v</pre> <p>displays the version and copyright message of dsedit.</p>
Usage	<ul style="list-style-type: none"> <li>• You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use dsedit.</li> <li>• You must set the DISPLAY environment variable before invoking dsedit, unless you are only using the -v parameter to display the version number.</li> <li>• For more information about the dsedit utility program, see Chapter 5, “Using dsedit” Also see the installation guide, and the configuration guide for your platform.</li> </ul>
See also	<p><b>Utilities</b> dscp</p>

## extractjava

**Description** Copies a retained JAR and the classes it contains from an Adaptive Server into a client file. Located in `$$SYBASE/$$SYBASE_OCS/bin`.

**In Windows** The utility is `extrjava.exe`, and is located in `%%SYBASE%\%%SYBASE_OCS%\bin`.

**Syntax** `extractjava` (`extrjava` in Windows)

```
-j jar_name
-f file_name
[-S server_name]
[-U user_name]
[-P password]
[-D database_name]
[-I interfaces_file]
[-a display_charset]
[-J client_charset]
[-z language]
[-t timeout]
[-v]
```

Or

```
extractjava -v
```

**Parameters**

`-j jar_name`

specifies the name assigned to the retained JAR in the database that is the source of the transfer.

`-f file_name`

specifies the name of the client file that is the target of the transfer.

`-S server_name`

specifies the name of the server.

`-U user_name`

specifies an Adaptive Server login name. If you omit the `-U` flag and parameter, or if you specify the `-U` flag with no parameter, Adaptive Server uses the current user's operating system login name.

`-P password`

specifies an Adaptive Server password. If you omit the `-P` flag and parameter, `extractjava` prompts for a password. If you specify the `-P` flag with no password, the null password is used.

`-D database_name`

specifies the name of the database in which to install the JAR. If you omit the `-D` flag, or if you specify the `-D` flag with no parameter, the user's default database is used.

- l *interfaces\_file***  
 specifies the name and location of the interfaces file to search when connecting to Adaptive Server. If you omit the **-l** flag and parameter, or if you specify the **-l** flag with no parameter, the interfaces file in the directory designated by your SYBASE environment variable is used.
- a *display\_charset***  
 allows you to use `extractjava` from a machine where the character set differs that of the server. Use **-a** in conjunction with **-J** to specify the character set translation file (*.xlt* file) required for the conversion. Use **-a** without **-J** only if the client character set is the same as the default character set.
- J *client\_charset***  
 specifies the character set to use on the client. `extractjava` uses a filter to convert input between *client\_charset* and the Adaptive Server character set.
- J *client\_charset*** requests that Adaptive Server convert to and from *client\_charset*, the character set used on the client.
- J** with no argument disables character set conversion. Use this if the client and server use the same character set.
- Omitting **-J** sets the character set to a default for the platform, which may not necessarily be the character set that the client is using. See the *System Administration Guide* for more information about character sets and associated flags.
- z *language***  
 specifies the name of an alternate language for displaying `extractjava` prompts and messages. Without the **-z** flag, `extractjava` uses the server's default language. You can add languages to an Adaptive Server during installation or afterward, using the `langinstall` utility or the `sp_addlanguage` stored procedure.
- t *timeout***  
 specifies the number of seconds before a SQL command times out. If you do not specify a timeout, the command runs indefinitely. This affects commands issued from within `extractjava`, not the connection time. The default timeout for logging into `extractjava` is 60 seconds.
- v**  
 prints the version number and copyright message for `extractjava` and then exits.

**Examples**

Downloads the classes associated with the employees JAR to the client file *newaddr.jar*.

- **UNIX:**

```
extractjava -j employees -f '/home/usera/jars/addr.jar' -new
```

- **Windows:**

```
extrjava -j employees -f '\\home\usera\jars\addr.jar' -new
```

Usage

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use extractjava.
- If the target client file already exists, extractjava overwrites its contents.
- The parameter flags -f, -j, -S, -U, -P, -D, and -I can be written with or without a space between the flag letter and the following parameter.
- When you execute extractjava, an exclusive lock is placed on sysxtypes.
- If -jar is specified, an exclusive table lock is placed on sysjars.

Permissions

You need to be a System Administrator or Database Owner to use extractjava.

Tables used

sysjars, sysxtypes

See also

**Commands** remove java

**Documentation** *Java in Adaptive Server Enterprise* for more information about how this utility is used when Java is enabled in the database.

**System procedures** sp\_helpjava

**Utilities** installjava

## installjava

Description	<p>Installs a JAR from a client file into an Adaptive Server. The utility is located in <code>\$\$SYBASE/\$\$SYBASE_OCS/bin</code>.</p> <p><b>Windows</b> The utility is <code>instjava.exe</code>, located in <code>\$\$SYBASE%\\$SYBASE_OCS%bin</code>.</p>
Syntax	<pre>installjava   -f <i>file_name</i>   [ -new   -update ]   [ -j <i>jar_name</i> ]   [ -S <i>server_name</i> ]   [ -U <i>user_name</i> ]   [ -P <i>password</i> ]   [ -D <i>database_name</i> ]   [ -I <i>interfaces_file</i> ]   [ -a <i>display_charset</i> ]   [ -J <i>client_charset</i> ]   [ -z <i>language</i> ]   [ -t <i>timeout</i> ]   [-v]</pre> <p>Or</p> <pre>installjava -v</pre>
Parameters	<p><b>-f <i>file_name</i></b> is the name of the source file containing the classes to be installed in the database.</p> <p><b>-new   -update</b> specifies whether the classes in the file already exist in the database. If you specify:</p> <ul style="list-style-type: none"> <li>• <b>-new</b> – you cannot install a class with the same name as an existing class</li> <li>• <b>-update</b> – you can install a class with the same name as an existing class, and the newly installed class replaces the existing class</li> </ul> <p><b>-j <i>jar_name</i></b> is the name of the JAR containing the classes to be installed in the database. Indicates that the JAR file should be saved in the database and associated with the classes it contains.</p> <p><b>-S <i>server_name</i></b> is the name of the server.</p> <p><b>-U <i>user_name</i></b> is an Adaptive Server login name. If you omit the <b>-U</b> flag and parameter, or if you specify the <b>-U</b> flag with no parameter, .</p>

- P *password*  
is an Adaptive Server password. If you omit the -P flag and parameter, installjava prompts for a password. If you specify the -P flag with no password, the null password is used.
- D *database\_name*  
is the name of the database in which to install the JAR. If you omit the -D flag, or if you specify the -D flag with no parameter, the user's default database is used.
- I *interfaces\_file*  
is the name and location of the interfaces file to search when connecting to Adaptive Server. If you omit the -I flag and parameter, or if you specify the -I flag with no parameter, the interfaces file in the directory designated by your SYBASE environment variable is used.
- a *display\_charset*  
allows you to use installjava from a machine where the character set differs that of the server. Use -a in conjunction with -J to specify the character set translation file (.xlt file) required for the conversion. Use -a without -J only if the client character set is the same as the default character set.
- J *client\_charset*  
specifies the character set to use on the client. installjava uses a filter to convert input between *client\_charset* and the Adaptive Server character set.
  - J *client\_charset* requests that Adaptive Server convert to and from *client\_charset*, the character set used on the client.
  - J with no argument disables character set conversion. Use this if the client and server use the same character set.

Omitting -J sets the character set to a default for the platform, which may not necessarily be the character set that the client is using. See the *System Administration Guide* for more information about character sets and associated flags.
- z *language*  
is the name of an alternate language for displaying installjava prompts and messages. Without the -z flag, installjava uses the server's default language. You can add languages to an Adaptive Server during installation or afterward, using the langinstall utility or the sp\_addlanguage stored procedure.



**-t *timeout***

specifies the number of seconds before a SQL command times out. If you do not specify a timeout, the command runs indefinitely. This affects commands issued from within `installjava`, not the connection time. The default timeout for logging into `installjava` is 60 seconds.

**-v**

prints the version number and copyright message for `installjava` and then exits.

**Examples**

**Example 1** Installs `addr.jar` and its classes, but does not retain the association between the JAR and classes:

```
installjava -f '/home/usera/jars/addr.jar' -new
```

**In Windows:**

```
instjava -f '\\home\usera\jars\addr.jar' -new
```

**Example 2** Reinstalls `addr.jar` and associates its classes with the employees JAR name:

```
installjava -f '/home/usera/jars/addr.jar' -update -j employees
```

**In Windows:**

```
instjava -f '\\home\usera\jars\addr.jar' -update -j employees
```

**Usage**

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use `installjava`.
- Any user can reference installed classes.
- The parameter flags `-f`, `-j`, `-S`, `-U`, `-P`, `-D`, and `-I` can be written with or without a space between the flag letter and the following parameter.

**Adding new JARs**

- If you use `new` with the `-jar` option and a JAR of that name already exists in the database, an exception is raised.
- If any classes of the same name as those in the source JAR already exist in the database, an exception is raised.

## Updating JARs and classes

---

**Warning!** If you alter a class used as a column datatype by reinstalling a modified version of the class, you must make sure that the modified class can read and use existing objects (rows) in tables using that class as a datatype. Otherwise, you may be unable to access those objects without reinstalling the class.

---

- If you use `-update` with the `-jar` option:
  - All classes in the database associated with the target JAR are deleted from the database and the classes in the source JAR file installed in their place.
  - If a class in the source JAR file is already installed in the database but is not attached to a JAR, the class in the source JAR is installed in the database and the unattached class is deleted.
- If you use `-update` without the `-jar` option:
  - Classes in the source JAR file replace unattached classes of the same name.
  - Classes in the source JAR that do not correspond to an installed class are installed as unattached classes in the database.
- If you install a new JAR with a replacement for an installed class that is referenced by a SQLJ procedure or function, make sure that the newly installed class has a valid signature for the SQLJ routine. If the signature is invalid, an exception is raised when the SQLJ routine is invoked.

### Locks

- When you execute `installjava`, an exclusive lock is placed on `sysxtypes`.
- If `-jar` is specified, an exclusive table lock is placed on `sysjars`.

Permissions

You need to be a System Administrator or Database Owner to use `installjava`.

Tables used

`sysjars`, `sysxtypes`

See also

**Commands** `remove java`

**Documentation** *Java in Adaptive Server Enterprise* for more information about how this utility is used when Java is enabled in the database.

**System procedures** `sp_helpjava`

**Utilities** `extractjava`

## isql

**Description** Interactive SQL parser to Adaptive Server. Located in `$$SYBASE/$SYBASE_OCS/bin`.

**Windows** The utility is `isql.exe`, located in `$$SYBASE%\$$SYBASE_OCS%\bin`.

**Syntax** `isql [-b] [-e] [-F] [-p] [-n] [-v] [-X] [-Y] [-Q]`  
`[-a display_charset]`  
`[-A packet_size]`  
`[-c cmdend]`  
`[-D database]`  
`[-E editor]`  
`[-h header]`  
`[-H hostname]`  
`[-i inputfile]`  
`[-I interfaces_file]`  
`[-J client_charset]`  
`[-K keytab_file]`  
`[-l login_timeout]`  
`[-m errorlevel]`  
`[-o outputfile]`  
`[-P password]`  
`[-R remote_server_principal]`  
`[-s colseparator]`  
`[-S server_name]`  
`[-t timeout]`  
`-U username`  
`[-V [security_options]]`  
`[-w columnwidth]`  
`[-z locale_name]`  
`[-Z security_mechanism]`  
 `[--conceal]`

---

**Note** See Table 7-2 for a list and description of commands that can be executed from within `isql`.

---

**Parameters**

- `-b`  
disables the display of the table headers output.
- `-e`  
echoes input.

- F  
enables the FIPS flagger. When you specify the -F parameter, the server returns a message when it encounters a non-standard SQL command. This option does not disable SQL extensions. Processing completes when you issue the non-ANSI SQL command.
- p  
prints performance statistics.
- n  
removes numbering and the prompt symbol (>) from the echoed input lines in the output file when used in conjunction with -e.
- v  
prints the version number and copyright message for isql and then exits.
- X  
initiates the login connection to the server with client-side password encryption. isql (the client) specifies to the server that password encryption is desired. The server sends back an encryption key, which isql uses to encrypt your password, and the server uses the key to authenticate your password when it arrives.  
  
If isql crashes, the system creates a core file that contains your password. If you did not use the encryption option, the password appears in plain text in the file. If you used the encryption option, your password is not readable.
- Y  
tells the Adaptive Server to use chained transactions.
- Q  
provides clients with failover property. See *Using Sybase Failover in a High Availability System* for more information.

**-a** *display\_charset*

runs isql from a terminal whose character set differs from that of the machine on which isql is running. Use -a in conjunction with -J to specify the character set translation file (*.xlt* file) required for the conversion. Use -a without -J only if the client character set is the same as the default character set.

---

**Note** The *ascii\_7* character set is compatible with all character sets. If either the Adaptive Server character set or the client character set is set to *ascii\_7*, any 7-bit ASCII character can pass unaltered between client and server. Other characters produce conversion errors. For more information on character set conversion, see the *System Administration Guide*.

---

**-A** *packet\_size*

specifies the network packet size to use for this isql session. For example, the following sets the packet size to 4096 bytes for this isql session:

```
isql -A 4096
```

- To check your network packet size, enter:

```
select * from sysprocesses
```
- The value is displayed under the *network\_pktsz* heading.
- *size* must be between the values of the default network packet size and maximum network packet size configuration parameters, and must be a multiple of 512.
- Use larger-than-default packet sizes to perform I/O-intensive operations, such as *readtext* or *writetext* operations.
- Setting or changing Adaptive Server's packet size does not affect the packet size of remote procedure calls.

**-c** *cmdend*

changes the command terminator. By default, you terminate commands and send them to by typing "go" on a line by itself. When you change the command terminator, do not use SQL reserved words or control characters.

**-D** *database*

selects the database in which the isql session begins.

**-E** *editor*

specifies an editor other than the default editor *vi*. To invoke the editor, enter its name as the first word of a line in isql.

- h *headers***  
specifies the number of rows to print between column headings. The default prints headings only once for each set of query results.
- H *hostname***  
sets the client host name.
- i *inputfile***  
specifies the name of the operating system file to use for input to isql. The file must contain command terminators (“go” is the default).
- Specifying the parameter as follows is equivalent to *< inputfile*:  
`-i inputfile`
  - If you use *-i* and do not specify your password on the command line, isql prompts you for it.
  - If you use *< inputfile* and do not specify your password on the command line, you must specify your password as the first line of the input file.
- l *interfaces\_file***  
specifies the name and location of the interfaces file to search when connecting to Adaptive Server. If you do not specify *-l*, isql looks for a file named *interfaces* in the directory specified by your SYBASE environment variable.
- J *client\_charset***  
specifies the character set to use on the client. *-J client\_charset* requests that Adaptive Server convert to and from *client\_charset*, the character set used on the client. A filter converts input between *client\_charset* and the Adaptive Server character set.
- J* with no argument sets character set conversion to NULL. No conversion takes place. Use this if the client and server use the same character set.
- Omitting *-J* sets the character set to a default for the platform. The default may not necessarily be the character set that the client is using. For more information about character sets and the associated flags, see “Configuring Client/Server Character Set Conversions,” in the *System Administration Guide, Volume One*.
- K *keytab\_file***  
specifies the path to the keytab file used for authentication in DCE.

- l *login\_timeout***  
specifies the maximum timeout value allowed when connecting to Adaptive Server. The default is 60 seconds. This value affects only the time that isql waits for the server to respond to a login attempt. To specify a timeout period for command processing, use the **-t *timeout*** parameter.
- m *errorlevel***  
customizes the error message display. For errors of the severity level specified or higher, only the message number, state, and error level are displayed; no error text appears. For error levels lower than the specified level, nothing appears.
- o *outputfile***  
specifies the name of an operating system file to store the output from isql. Specifying the parameter as **-o *outputfile*** is similar to **> *outputfile***
- P *password***  
specifies your Adaptive Server password. If you do not specify the **-P** flag, isql prompts for a password. If your password is NULL, use the **-P** flag without any password.
- R *remote\_server\_principal***  
specifies the principal name for the server as defined to the security mechanism. By default, a server's principal name matches the server's network name (which is specified with the **-S** parameter or the DSQUERY environment variable). Use the **-R** parameter when the server's principal name and network name are not the same.
- s *colseparator***  
resets the column separator character, which is blank by default. To use characters that have special meaning to the operating system (for example, "|", ",", "&", "<", ">"), enclose them in quotes or precede them with a backslash.

The column separator appears at the beginning and the end of each column of each row.
- S *server\_name***  
specifies the name of the Adaptive Server to which to connect. isql looks this name up in the interfaces file. If you specify **-S** with no argument, isql looks for a server named SYBASE. If you do not specify **-S**, isql looks for the server specified by your DSQUERY environment variable.

**-t *timeout***

specifies the number of seconds before a SQL command times out. If you do not specify a timeout, the command runs indefinitely. This affects commands issued from within isql, not the connection time. The default timeout for logging into isql is 60 seconds.

**-U *username***

specifies a login name. Login names are case sensitive.

**-V *security\_options***

specifies network-based user authentication. With this option, the user must log in to the network's security system before running the utility. In this case, users must supply their network user name with the -U option; any password supplied with the -P option is ignored.

You can follow -V with a *security\_options* string of key-letter options to enable additional security services. These key letters are:

- c – Enable data confidentiality service
- i – Enable data integrity service
- m – Enable mutual authentication for connection establishment
- o – Enable data origin stamping service
- q – Enable out-of-sequence detection
- r – Enable data replay detection

**-w *columnwidth***

sets the screen width for output. The default is 80 characters. When an output line reaches its maximum screen width, it breaks into multiple lines.

**-z *locale\_name***

specifies the official name of an alternate language to display isql prompts and messages. Without -z, isql uses the server's default language. You can add languages to an Adaptive Server during installation or afterward, using the langinstall utility (langinst in Windows) or the sp\_addlanguage stored procedure.

**-Z *security\_mechanism***

specifies the name of a security mechanism to use on the connection.

Security mechanism names are defined in the *libtcl.cfg* configuration file located in the *ini* subdirectory below the Sybase installation directory. If no *security\_mechanism* name is supplied, the default mechanism is used. For more information on security mechanism names, see the description of the *libtcl.cfg* file in the *Open Client and Open Server Configuration Guide*.



--conceal

Hides your input during an isql session. The --conceal option is useful when entering sensitive information, such as passwords. The syntax for the --conceal option is:

```
isql --conceal [':?' | 'wildcard']
```

The maximum length of *wildcard*, a 32-byte variable, is 80 characters.

---

**Note** --conceal is silently ignored in batch mode.

---

## Examples

**Example 1** Puts you in a text file where you can edit the query. When you write and save the file, you are returned to isql. The query appears; type “go” on a line by itself to execute it:

```
isql -Ujoe -Pabracadabra
1> select *
2> from authors
3> where city = "Oakland"
4> vi
```

**Example 2** reset clears the query buffer. quit returns you to the operating system:

```
isql -Ualma
Password:
1> select *
2> from authors
3> where city = "Oakland"
4> reset
1> quit
```

**Example 3** Specifies that you are running isql from a Macintosh against a server that is using the roman8 character set:

```
isql -a mac -J roman8
```

**Example 4** Creates column separators using the “#” character in the output in the pubs2 database for store ID 7896:

```
isql -Usa -P -s#
1> use pubs2
2> go
1> select * from sales where stor_id = "7896"
#stor_id#ord_num          #date          #
#-----#-----#-----#
#7896   #124152          #      Aug 14 1986 12:00AM#
#7896   #234518          #      Feb 14 1991 12:00AM#
```

(2 rows affected)

**Example 5** Changes password without displaying the password entered. This example uses “old” and “new” as prompt labels:

```
$ isql -Uguest -Pguest -Smyase --conceal
sp_password
:?: old
'
:?:? new
-----
old
new
Confirm new
Password correctly set.
```

(Return status 0)

**Example 6** Activates a role for the current user. This example uses a custom wildcard and the prompt labels “role” and “password:”

```
$ isql -UmyAccount --conceal '*'
Password:
set role
* role
with passwd
** password
on
go

role
password
Confirm password
```

**Example 7** Changes password without displaying the password entered. This example uses the default wildcard as the prompt label:

#### Usage

- Use this syntax for isql\_r if you are using threaded drivers.
- Use this syntax for isql if you are using threaded drivers in the IBM platform.
- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use isql.
- Table 7-2 describes the commands you can execute from within interactive isql.

**Table 7-2: Commands available within isql**

Command	Description
<code>:r filename</code>	Reads an operating system file into the command buffer. Do not include the command terminator in the file; once you have finished editing, enter the terminator interactively on a line by itself.
<code>:R filename</code>	Reads an operating system file into the command buffer and then displays it. Do not include the command terminator in the file; once you have finished editing, enter the terminator interactively on a line by itself.
<code>use database_name</code>	Changes the current database.
<code>!! os_command</code>	Executes an operating system command. Place at the start of a line.
<code>&gt; file_name</code>	Redirects the output of the T-SQL command to <i>file_name</i> . The following example inserts the server version into <i>file_name</i> : <pre>select @@version go &gt; file_name</pre>
<code>&gt;&gt; file_name</code>	Appends the output of the T-SQL command to <i>file_name</i> . The following example appends the server version to <i>file_name</i> : <pre>select @@version go &gt;&gt; file_name</pre>
<code>  command</code>	Pipes the output of the T-SQL command to an external command. The following example finds all instances of “sa” in the listing produced by <code>sp_who</code> : <pre>sp_who go   grep sa</pre>
<code>vi</code> (UNIX) <code>edit</code> (Windows)	Calls the default editor.
<code>reset</code>	Clears the query buffer.
<code>quit</code> or <code>exit</code>	Exits isql.

- The 5701 (“changed database”) server message is no longer displayed after login or issuing a `use database` command.
- Error message format differs from earlier versions of isql. If you have scripts that perform routines based on the values of these messages you may need to rewrite them.
- To use isql interactively, give the command `isql` (and any of the optional parameters) at your operating system prompt. The isql program accepts SQL commands and sends them to Adaptive Server. The results are formatted and printed on standard output. Exit isql with `quit` or `exit`.

- Terminate a command by typing a line beginning with the default command terminator `go` or another command terminator, if the `-c` parameter is used. You can follow the command terminator with an integer to specify how many times to run the command. For example, to execute this command 100 times, type:

```
select x = 1
go 100
```

The results display once at the end of execution.

- If you enter an option more than once on the command line, `isql` uses the last value. For example, if you enter the following command, “send”, the second value for `-c`, overrides “.”, the first value:

```
isql -c "." -c send
```

This enables you to override any aliases you set up.

- To call an editor on the current query buffer, enter its name as the first word on a line. Define your preferred callable editor by specifying it with the `EDITOR` environment variable. If `EDITOR` is not defined, the default is `vi` on UNIX and `edit` on Windows.

For example, if your `EDITOR` environment variable is set to “`emacs`,” then you must invoke it from within `isql` with “`emacs`” as the first word on the line.

- To clear the existing query buffer, type `reset` on a line by itself. `isql` discards any pending input. You can also press `Ctrl-c` anywhere on a line to cancel the current query and return to the `isql` prompt.
- Read in an operating system file containing a query for execution by `isql` as follows:

```
isql -U alma -Ppassword < input_file
```

The file must include a command terminator. The results appear on your terminal. Read in an operating system file containing a query and direct the results to another file as follows:

```
isql -U alma -Ppassword < input_file > output_file
```

- Case is significant for the `isql` flags.
- `isql` displays only six digits of float or real data after the decimal point, rounding off the remainder.

- You can include comments in a Transact-SQL statement submitted to Adaptive Server by isql. Open a comment with “/\*”. Close it with “\*/”, as shown in the following example:

```
select au_lname, au_fname
/*retrieve authors' last and first names*/
from authors, titles, titleauthor
where authors.au_id = titleauthor.au_id
and titles.title_id = titleauthor.title_id
/*this is a three-way join that links authors
**to the books they have written.**/
```

If you want to comment out a go command, it should not be at the beginning of a line. For example, use the following to comment out the go command:

```
/*
**go
*/
```

Do not use the following:

```
/*
go
*/
```

- isql defines the order of the date format as month, date, and year (mm dd yyyy hh:mmAM (or PM)) regardless of the locale environment. To change this default order, use the convert function.
- In an isql session, the default prompt label is either the default wildcard :? or the value of *wildcard*. You can customize the prompt label by providing a one-word character string, with a maximum length of 80 characters after a wildcard. If you specify a prompt label that is more than one word, the characters after the first word are ignored.

---

**Note** In an isql session, isql recognizes :?, or the value of *wildcard*, as wildcards only when these characters are placed at the beginning of an isql line.

---

See also

**Commands** create schema, set

**Datatype** exact numeric datatypes

**Documentation** Chapter 2, “Using Interactive isql from the Command Line” for details on isql, and the *Reference Manual* for more information regarding default network packet size and maximum network packet size configuration parameters.

**Functions** convert

**System ESP** xp\_sendmail

**System procedures** sp\_addlanguage, sp\_addlogin, sp\_addremotelogin, sp\_add\_resource\_limit, sp\_bindexclass, sp\_configure, sp\_defaultlanguage, sp\_droplanguage, sp\_helplanguage, sp\_processmail, sp\_remoteoption, sp\_serveroption, sp\_showcontrolinfo, sp\_unbindexclass, sp\_volchanged

# langinstall

Description	<p>Installs a new language in an Adaptive Server. Located in <code>\$\$SYBASE/\$SYBASE_ASE/bin</code>.</p> <p><b>Windows</b> The utility is <code>langinst.exe</code>, located in <code>\$\$SYBASE%\\$SYBASE_ASE%bin</code>.</p>
Syntax	<pre>langinstall   [-S <i>server</i>]   [-U <i>user</i>]   [-P <i>password</i>]   [-R <i>release_number</i>]   [-I <i>path</i>]   <i>language</i>   <i>character_set</i></pre> <p>Or</p> <pre>langinstall -v</pre>
Parameters	<p><b>-S <i>server</i></b> specifies the name of the Adaptive Server to which to connect. If you do not specify <b>-S</b>, <code>langinstall</code> uses the server specified by your <code>DSQUERY</code> environment variable. If <code>DSQUERY</code> is not set, <code>langinstall</code> attempts to connect to a server named SYBASE.</p> <p><b>-U <i>user</i></b> specifies a login name. Login names are case sensitive.</p> <p><b>-P <i>password</i></b> specifies the System Administrator's ("sa" account) password. If you omit <b>-P</b>, <code>langinstall</code> prompts for the "sa" account password.</p> <p><b>-R <i>release_number</i></b> specifies the release number, in the format <i>n.n.n</i>, to use to upgrade messages in <code>master.sysmessages</code>. Use <b>-R</b> only in failure conditions, such as if <code>langinstall</code> (<code>langinst</code> in Windows) fails, in case of user error, or when you think that messages in <code>sysmessages</code> are out of date.</p> <p>The <b>-R</b> parameter forces <code>langinstall</code> to collect messages from a release previous to the current one. <code>langinstall</code> compares the existing messages with the ones to be installed and replaces any that have changed.</p> <p>For example, if the current version is 15.0, and the previous version was 12.5, and you think <code>sysmessages</code> may not be correct, include the messages from the earlier version in the <code>syslanguages.upgrade</code> column (12.5 in this case) by specifying <b>-R 12.5</b>. <code>langinstall</code> then installs all messages from Adaptive Server 12.5.</p>

**-l path**

specifies the name and location of the interfaces file (*sql.ini* file in Windows) that langinstall searches when connecting to Adaptive Server. If you do not specify -l, langinstall uses the interfaces file in the directory specified by the SYBASE environment variable. If SYBASE is not set, langinstall looks for the default *SYBASE* directory.

**language**

is the official name of the language to be installed. You must specify a language.

**character\_set**

is the name of Adaptive Server's default character set. *character\_set* indicates the directory name of the localization files for the language. The *common.loc* and *server.loc* localization files for an official language reside in the character set directory *\$SYBASE/locales/language/character\_set* in UNIX platforms, or *%SYBASE%\locales\language\character\_set* in Windows. You must specify a character set.

**-v**

prints the version number and copyright message for langinstall and then exits.

Usage

- The Adaptive Server installation program runs langinstall automatically for a new installation as well as for customers who are upgrading from an earlier version.
- langinstall does the following:
  - Adds the specified language-specific information to master..syslanguages using sp\_addlanguage. If the language already exists, langinstall updates the appropriate row in syslanguages.
  - Adds to, updates, and deletes error messages as necessary from master..sysmessages.
  - Updates syslanguages.update, inserting the new release number.
- langinstall validates the entries in the localization file sections that it uses. If anything is missing, langinstall prints an error message and does not add the language to syslanguages.
- langinstall compares the version numbers of each localization file it uses, *common.loc* and *server.loc*. If they are not the same, it prints a warning message. syslanguages.upgrade is always set according to the version number in *server.loc*.

Permissions

Only a System Administrator using the "sa" account can run langinstall.





## optdiag

Description	<p>Displays optimizer statistics or loads updated statistics into system tables. optdiag is located in <code>\$\$SYBASE/\$\$SYBASE_ASE/bin</code>.</p> <p><b>Windows</b> The utility is <code>optdiag.exe</code>, located in <code>\$\$SYBASE%\\%\$SYBASE_ASE%\\bin</code>.</p>
Syntax	<pre>optdiag [binary] [simulate] statistics         { -i <i>input_file</i>   <i>database</i>.[<i>owner</i>].[<i>table</i>.[<i>column</i>] ] } [-o <i>output_file</i>] }         [-U <i>user_name</i>]         [-P <i>password</i>]         [-T <i>trace_value</i>]         [-I <i>interfaces_file</i>]         [-S <i>server</i>]         [-v]         [-h]         [-s]         [-z <i>language</i>]         [-J <i>client_character_set</i>]         [-a <i>display_charset</i>]</pre>
Parameters	<p><b>binary</b> extracts statistics in human-readable form and in binary form. When used with an input file (<code>-i <i>input_file</i></code>), loads binary statistics into system tables.</p> <p><b>simulate</b> specifies that optdiag display or load simulated statistics. See the <i>Performance and Tuning Guide</i>.</p> <p><b>-i <i>input_file</i></b> specifies the name of the operating system file to use for optdiag input. Specifying an input file causes optdiag to update optimizer statistics for the table or column by using the values in the specified file (also called “input mode”).</p> <p><b><i>database</i></b> is the name of the database whose statistics you want displayed. In input mode, optdiag uses the database name as specified in the file, and does not accept a database name from the command line.</p> <p><b><i>owner</i></b> is the name of a table owner.</p> <ul style="list-style-type: none"><li>• In display mode, if you do not specify an owner, but do specify a table name, optdiag displays output for all of the owners of a table.</li><li>• In input mode, optdiag ignores the table owner specified on the command line and uses the value in the input file.</li></ul>

*table*

is the name of the table to survey for statistics.

- If the command does not include an owner name or a table name, `optdiag` displays statistics for all tables in the database.
- If the command includes an owner name, but no table name, `optdiag` displays all of the tables that belong to the specified owner.
- In input mode, `optdiag` ignores the table name specified on the command line and uses the value from the input file.

*column*

is the name of the column to survey.

- If the command does not include a column name, `optdiag` displays all statistics for a table.
- In input mode, `optdiag` ignores the column name on the command line and uses the values from the input file.

`-o output_file`

specifies the name of an operating system file to store the output from `optdiag`. If a file with the same name already exists, `optdiag` overwrites that file without warning.

`-U user_name`

specifies an Adaptive Server login name.

`-P password`

specifies your Adaptive Server password. If you do not specify the `-P` flag, `optdiag` prompts for a password.

`-T trace_value`

sets trace flags for the `optdiag` session. The `optdiag` trace flags are:

Flag value	Meaning
1	Do not stop with a warning if the <code>optdiag</code> version of Adaptive Server in use does not match the Adaptive Server version in the input file.
2	Display status message "Next table is <i>table_name</i> " when in input mode.
4	Skip consistency checking for step numbers while loading histograms in input mode.
6	Display lines of input file during input mode. This flag has no effect in display mode.
7	Do not stop with a warning if the <code>optdiag</code> input file does not include sampling percent information.

**-l *interfaces\_file***

specifies the name and location of the interfaces file to use when connecting to Adaptive Server.

If you do not use **-l** and specify an interfaces file name, **optdiag** looks for the interfaces file (*interfaces* in UNIX), in the directory specified by the SYBASE environment variable. In Windows, **optdiag** looks for a file named *sql.ini* in the *ini* subdirectory in the Sybase installation directory (*d:\sybase*). Then, if SYBASE is not set, **optdiag** looks for the file in the default *SYBASE* directory (*%SYBASE%* in Windows).

**-S *server***

specifies the name of the Adaptive Server to which to connect. **optdiag** looks for this name in the *interfaces* file (*sql.ini* in Windows).

- If you use **-S** without specifying a server name, **optdiag** looks for a server named SYBASE.
- When you do not use **-S**, **optdiag** looks for the server that your DSQUERY environment variable specifies.

**-v**

displays the version number of and a copyright message for **optdiag** and exits.

**-h**

displays the **optdiag** syntax help.

**-s**

includes system tables in **optdiag** output. By default, only user tables are included.

**-z *language***

is the official name of an alternate language that the server uses both for date formats and to display **optdiag** prompts and messages. Without the **-z** flag, **optdiag** uses the server's default language.

You can add languages to Adaptive Server either during or after installation. After Adaptive Server installation, use either the *langinstall* utility or the *sp\_addlanguage* stored procedure to add a language.

**-J *client\_charset***

specifies the character set to use on the client. A filter converts input between *client\_charset* and the Adaptive Server character set.

By using **-J *client\_charset***, you request that Adaptive Server convert data to and from *client\_charset*, the client's character set.

By using **-J** without a character set name, you specify character set conversion as NULL; no conversion takes place. Use this **-J** alone when the client and server are using the same character set.

By omitting **-J**, you set the character set to the default set for the platform. A filter converts input between the default set and the Adaptive Server character set. Keep in mind that the default may not necessarily be the character set that the client is using.

For more information about character sets and their associated flags, see the *System Administration Guide*.

**-a display\_charset**

runs optdiag from a terminal with a character set that differs from that of the machine on which optdiag is running.

- Use **-a** in conjunction with **-J** to specify the character set translation (*.xlt*) file required for the conversion.
- Use **-a** without **-J** only if the client character set is the same as the default character set.

---

**Note** The `ascii_7` character set is compatible with all character sets. If either the Adaptive Server character set or the client character set is set to `ascii_7`, any 7-bit ASCII character can pass unaltered between client and server. Any other characters produce conversion errors. For more information on character-set conversion, see the *System Administration Guide*.

---

---

**Note** On some Linux platforms, the `LANG` environment variable might be set by default to “`en_US.UTF-8`,” which can cause unnecessary `LONGCHAR` conversion between the client and server. If your server and client have different charsets, Sybase recommends that you bypass the conversion using one of these methods:

- `unsetenv LANG`
  - `setenv LANG C`
  - `optdiag -J`
  - `optdiag -Jiso-1` (if your server uses iso-1)
- 

**Examples**

**Example 1** Displays statistics for all user tables in the `pubs2` database and places the output in the *pubs2.opt* file:

```
optdiag statistics pubs2 -Usa -Ppasswd -o pubs2.opt
```

**Example 2** Displays statistics for the `titles` table:

```
optdiag statistics pubs2..titles -Usa -Ppasswd -o titles.opt
```

**Example 3** Displays statistics using the `roman8` character set and row labels and error messages in French:

```
optdiag statistics pubs2..titles -Usa -Ppasswd -o titles.opt -J roman8  
-z french
```

**Example 4** Displays binary statistics for the `price` column in the `titles` table:

```
optdiag binary statistics pubs2..titles.price -Usa -Ppasswd -o price.opt
```

**Example 5** Loads edited statistics from the *price.opt* file:

```
optdiag statistics -i price.opt -Usa -Ppasswd
```

Usage

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use optdiag.
- By default, optdiag does not include the system tables when you display statistics for a database. To include the system tables in the output, use the -s flag.
- You cannot specify a particular partition on the optdiag command line; optdiag displays statistics for all partitions of a specified table.
- When you use binary mode, optdiag displays the human-readable values with comment marks (#) at the beginning of the lines, as shown in this example:

```
Statistics for column:           "price"
Last update of column statistics: Jan 20
1998   7:16PM
Statistics loaded from Optdiag.
Range cell density:             0x3f8b9cfefece26bf
# Range cell density:           0.0134830400000000
Total density:                  0x3f8b9cfefece26bf
# Total density:                0.0134830400000000
Range selectivity:              default used (0.33)
# Range selectivity:            default used (0.33)
In between selectivity:         default used (0.25)
# In between selectivity:       default used (0.25)
```

- When you use optdiag with an input file to change statistics, it ignores all characters after the “#” in a line.
- Converting floating-point values may lead to rounding errors when you use files for input.

When you are loading statistics on the same hardware platform, edit the statistics using the binary values to provide greater precision.

- optdiag displays the statistic, sampling percent last used, which indicates that statistics are gathered with a user-specified sampling percent.
- optdiag displays statistics for each partition of a multi-partitioned table or index.

- optdiag displays global- and partition-level statistics for each column in a table with multiple partitions.

#### Byte ordering and binary optdiag files

- Do not use the binary mode option to move statistics between Adaptive Servers on machines that use different byte ordering.
  - On an incompatible architecture server, always comment out binary statistics and load the human-readable statistics.
  - On a compatible architecture server, you can load either binary statistics or human-readable statistics.

#### Input mode

- When you use the `-i input_file` syntax, optdiag reads the file as named and updates statistics in sysstatistics.
- optdiag input mode changes the allow update to system tables configuration parameter by setting the parameter to 1 at the beginning of the session, and then to 0 at the end of the session.
- During histogram input, the process checks the following rules and displays error messages for any violated rules:
  - The step numbers must increase monotonically, unless the command includes the `-T4` trace flag.
  - The column values for the steps must increase monotonically.
  - The weight for each cell must be between 0.0 and 1.0.
  - The total of weights for a column must be close to 1.0.
  - The first cell represents null values, and it must be present, even in columns that do not allow null values. There must be only one cell to represent the null value.
  - Two adjacent cells must not both use the `<` (less than) operator.

See also

**Commands** create index, delete statistics, set, update statistics

**Documentation** *Performance and Tuning Guide* for more information about the optdiag command and an explanation of the optdiag output, and *Performance and Tuning Guide* for more information on changing statistics using optdiag.

**System procedures** sp\_addlogin, sp\_configure, sp\_defaultlanguage, sp\_droplanguage, sp\_flushstats, sp\_helplanguage



## preupgrade

Description	<p>Performs tests on an installation or database to determine its readiness for upgrade, and reports found problems. <code>preupgrade</code> is located in <code>\$\$SYBASE/\$SYBASE_ASE/upgrade</code>.</p> <p><b>Windows</b> The utility is <code>preupgrd.exe</code>, located in <code>\$\$SYBASE%\\$\$SYBASE_ASE%\upgrade</code>.</p>
Syntax	<pre>preupgrade [-v] [-h] [-N]            [-p <i>skip_sybprocs</i>]            [-D <i>database_name</i>]            [-I <i>interfaces_file</i>]            [-P <i>password</i>]            [-S <i>server_name</i>]            [-U <i>user_name</i>]            [-X <i>option[,option]...</i>]</pre>
Parameters	<p><code>-D <i>database_name</i></code> limits checking to the named database and a subset of possible checks. Use this parameter to check newly loaded databases before bringing them online.</p> <p><code>-h</code> prints help text and then exits.</p> <p><code>-I <i>interfaces_file</i></code> specifies an interfaces file for the server. The default is <code>\$\$SYBASE/interfaces</code>.</p> <p><code>-N</code> specifies <code>preupgrade</code> is to run in noninteractive mode. Thus, if <code>preupgrade</code> determines that any database is too small, the utility exits immediately.</p> <p><code>-p <i>skip_sybprocs</i></code> specifies whether you want to skip the parsing validity test on stored procedures.</p> <p>During the upgrade process, stored procedures are automatically re-created internally from the text source, requiring that they be parsed correctly. The valid options for <code>preupgrade -p</code> are:</p> <ul style="list-style-type: none"> <li>• <code>-p</code> – skips the parsing in all databases</li> <li>• <code>-p skip_procs</code> – skips parsing in sybssystemprocs while parsing the other databases</li> <li>• Not using <code>-p</code> – parses text everywhere. This is the default.</li> </ul>

- P** *password*  
specifies the password for connecting to the server. Sybase recommends that you do not use this option on the command line as the password is then visible to other users when they view displays of system information. Rather, wait until Adaptive Server prompts for a password, and enter the information then.
- S** *server\_name*  
specifies the name of the server to which you want to connect. This server must be listed in the interfaces file specified by the **-I** parameter. The default is `$DSQUERY`.
- U** *user\_name*  
specifies the user name to use when connecting to the server. The default is "sa." *user\_name* must have "sa\_role" privileges on the server being checked.

---

**Note** If you use the **-D** option, which limits checking to a named database, and that database is offline, you must enter "sa" or accept the default as the user name.

---

- v**  
prints version information and exits.
- X** *option[, option...]*  
specifies a list of checks to be made. The default is all checks, except when using the **-D** option, which uses only a subset of available checks. If you specify the **-X** option more than once on the command line, preupgrade performs only those checks in the last entered **-X** parameter.

When using the **-X** parameter with an options list, you must either:

- List options without a space between the comma and the next option, or
- Surround the options list with quotes.

Valid check options are described in Table 7-3. Sybase may occasionally change valid options for the **-X** parameter; use the **-h** parameter to view the current set of valid options

**Table 7-3: Check options with the -X parameter**

Check options	Options used with the -D parameter	Description
all		Performs all permitted checks. When used with the <b>-D</b> option, only checks subset of options. Otherwise, all options are checked.

Check options	Options used with the -D parameter	Description
cache		Checks the definition of default cache size. If the definition is DEFAULT, enter its current value in the configuration file as its actual value. This ensures that its size does not change because the new server's default value is different from the current server's default value.
config		Checks the server's configuration parameters to see if they are consistent with new requirements, and reports discrepancies.  Discrepancies can cause errors or warnings for certain parameters: <ul style="list-style-type: none"> <li>• Errors – occur when the current value of a parameter is outside the new server's range, or when its value is too low for upgrade.</li> <li>• Warnings – occur when the current value of a parameter is between the maximum and minimum values, but less than the default value</li> </ul>
data_mods		Performs updates to system tables, including clearing certain system table columns that are non-zero. Applies primarily to older Adaptive Servers, and will make no changes to newer systems.
datatype	X	Checks the systypes table to make sure that if existing datatypes use a system-defined name, type, or user type, they match what the new server expects. Reports discrepancies and suggests remedies.
db_size		Checks that certain system databases meet the minimum size requirements for installation.
free_space	X	Checks for free space in the named database or in all databases. Makes sure that there is sufficient free data and log space to perform the necessary upgrade steps.
object_id	X	Checks that object IDs of user-defined objects are not reserved for system objects. <ul style="list-style-type: none"> <li>• Adaptive Server 15.0 and later reserves objects IDs 1 – 255.</li> <li>• Adaptive Server 12.5.x and earlier reserves object IDs 1 – 99.</li> </ul> Does not issue an error. If you receive a warning that a user object ID is reserved, contact Sybase Technical support for directions for changing the user object ID after upgrade.
required_dbs		Checks that required system databases exist. Some versions of Adaptive Server may require specialized databases such as sybssystemdb.
sproc_text	X	Checks for stored procedure text in the named database or in all databases. After upgrade, you must recompile stored procedures from their source text. This check makes sure that all of the source text is both available and valid.
srvclass		Checks for servers classed as “generic” in master.dbo.syssservers. This class is deprecated by Adaptive Server 12.0 and later.
statistics		Checks for duplicate rows in sysstatistics. Duplicate rows may occur when upgrading from Adaptive Server 12.0 to Adaptive Server 15.0 and later due to schema changes in the sysstatistics table.

Examples

**Example 1** This example runs preupgrade and specifies checks for the default cache size, minimum database size, and duplicate rows in sysstatistics for the installation.

```
preupgrade -X cache,db_size,statistics
```

**Example 2** This example runs preupgrade to check a newly loaded, offline database. preupgrade checks datatypes, free space, object IDs, and stored procedure text.

```
preupgrade -Dmy_db -Usa
```

**Example 3** This example shows how the sqlupgrade utility uses preupgrade. When used in this way, preupgrade checks all databases, runs noninteractively, and exits with a failing status if any database is too small.

```
preupgrade -N
```

Usage

- When preupgrade finds no errors, it exits with status 0 (zero).
- preupgrade is primarily used before upgrading an installation to ensure the procedure will run smoothly. You can also run preupgrade after loading a database created by a previous version of Adaptive Server to check for problems that might prevent that database from upgrading while being brought online.
- sqlupgrade calls preupgrade as part of its normal procedure.
- Unless the -D parameter is included, preupgrade checks all databases in the system, and runs only on previous server versions.
- If the -D parameter is included, preupgrade:
  - Checks only the named database.
  - Can run be run against server of the same version number as the preupgrade utility.
  - Limits checking to a subset of possible checks. See Table 7-3 for a list of checks that are valid when used in conjunction with this parameter.
- The -D parameter is used primarily to check newly loaded databases before bringing them online. To do so, preupgrade must force access to offline databases, which requires that preupgrade log in as user “sa” The “sa\_role” privilege alone is insufficient. The default is -Usa.

When run as part of a normal upgrade, the -D parameter is optional, and you can choose a System Administrator login other than “sa.”

- When running preupgrade after upgrade to check a newly loaded but offline database, you must use the -D parameter and either specify the “sa” login or omit the -U parameter.
- When you specify the -D option, or specify a partial list of options with the -X parameter, preupgrade performs the specified checks and, if there are no errors, displays a list of checks performed in the exit message.
- You may want to use the -D and the -X parameters as part of the normal upgrade if preupgrade reports warnings or errors for a particular database or area. Using these parameters allows you to focus on problem areas without repeating unnecessary checks.

sqlupgrade always runs the full set of preupgrade checks. You can correct problems and run preupgrade to ensure that the problems are indeed corrected before letting sqlupgrade repeat the full preupgrade procedure.

**Permissions**

The user login specified by the -U parameter must have System Administrator privileges to run preupgrade on the server specified by the -S parameter.

When using the -D parameter to check an offline database, preupgrade must log in as user “sa.” The “sa\_role” privilege is insufficient

**See also**

**Utilities** sqlupgrade, sqlupgraderes

## pwdcrypt

**Description** Creates and prints an encrypted LDAP password in the *libtcl.cfg* file. pwdcrypt is located in *SYBASE/SYBASE\_OCS/bin*.

**Windows** The utility is located in *%SYBASE%\%SYBASE\_OCS%\bin*.

**Syntax** pwdcrypt

**Parameters** None

**Examples** Typing pwdcrypt at the prompt returns a request to enter your password twice, after which pwdcrypt returns the LDAP password:

```
pwdcrypt
Enter password please: password
Enter password again : password

The encrypted password:
0x01312a775ab9d5c71f99f05f7712d2cded288d0ae1ce79268d0e8669313d1bc4c706
```

Replace the last part of the LDAP URL in *libtcl.cfg* with this encrypted password:

```
ldap=libsybdldap.so
ldap://dolly:389/dc=sybase,dc=com????bindname=cn=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cded288d0ae1ce79268d0e8669313d1bc4c706
```

An unencrypted password looks like this:

```
ldap=libsybdldap.so
ldap://dolly:389/dc=sybase,dc=com????bindname=cn=Manager,dc=sybase,dc=com?
secret
```

**Usage** You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use pwdcrypt.

**Permissions** You must use file system permissions to prevent unauthorized access to this encrypted password in your *libtcl.cfg* file.

## showserver

**Description**                    **UNIX platforms only** Shows the Adaptive Servers and Backup Servers that are currently running on the local machine, available only in UNIX platforms. showserver is located in `$SYBASE/$SYBASE_ASE/install`.

**Syntax**                            showserver

**Parameters**                    None

**Examples**                        Shows the Adaptive Servers and Backup Servers that are currently running on the local machine:

```
showserver
```

```
USER          PID %CPU %MEM    SZ  RSS TT  STAT  START   TIME  COMMAND
user114276    0.0  1.7  712 1000 ?  S    Apr  5514:05  dataserver
-d greensrv.dat -sgreensrv -einstall/greensrv+_errorlog
sybase       1071  0.0  1.4  408  820 ?  S    Mar 28895:38
/usr/local/sybase/bin/dataserver -d/dev/rsd1f -e/install/errorlog
user128493    0.0  0.0 3692   0 ?  IW   Apr  1  0:10  backupserver -SSYB_BACKUP
-e/install/backup.log -Iinterfaces -Mbin/sybmultbuf -Lus_english -Jiso_1
```

**Usage**                            showserver displays process information about Adaptive Server or Backup Server. If no servers are running, only the header appears.

**See also**                        **Commands**    dataserver, startserver

**Function**    host\_name

**Utilities**    langinstall

## sqldbg

**Description** sqldbg is a command-line utility that debugs stored procedures and triggers. As with many source-level debuggers, you can:

- attach sqldbg to a task
- set, enable, and disable breakpoints
- step through a task one line at a time
- step into and out of procedures
- detach sqldbg from stored procedures or triggers once the debugging is complete.

**UNIX platforms** sqldbg is located in `$$SYBASE/$$SYBASE_ASE/bin`.

**Windows** sqldbg is located in `%SYBASE%\%SYBASE_ASE%\bin`.

---

**Note** You do not have the ability to view sqldbg version strings.

---

**Syntax** sqldbg  
           -U *username*  
           -P *password*  
           -S *host:port*

**Parameters**

- U *username*  
specifies the user name. You must insert a space between -U and *username*.
- P *password*  
specifies the user password. You must insert a space between -P and *password*.
- S *host:port*  
specifies the machine name and the port number. You must insert a space between -S and *host:port*.

**Examples** **Example 1** Shows sqldbg debugging stored procedures and triggers on host MERCURY:

```
$$SYBASE/$$SYBASE_ASE/bin/sqldbg -U sa -P -S MERCURY:16896
```

```
(sqldbg) stop in sp_who
Breakpoint moved to line 20
(sqldbg) run sp_who
(sp_who::20)if @@trancount = 0
(sqldbg) next
(sp_who::22) set chained off
(sqldbg) cont
```



```

fid spid status loginame origname hostname blk_spid dbname cmd block_xloid
0 2 sleeping NULL NULL 0 master NETWORK HANDLER 0
0 3 sleeping NULL NULL 0 master NETWORK HANDLER 0
0 4 sleeping NULL NULL 0 master DEADLOCK TUNE 0
0 5 sleeping NULL NULL 0 master MIRROR HANDLER 0
0 6 sleeping NULL NULL 0 master ASTC HANDLER 0
0 7 sleeping NULL NULL 0 master ASTC HANDLER 0
0 8 sleeping NULL NULL 0 master CHECKPOINT SLEEP 0
0 9 sleeping NULL NULL 0 master HOUSEKEEPER 0
0 10 running sa sa 0 master SELECT 0
0 11 sleeping sa sa
(sqldbg) show breakpoints
1 stop in sp_who
(sqldbg)

```

**Example 2** In this example, the System Administrator first logs in to Adaptive Server using isql, then starts sqldbgr from the command line to debug a stored procedure that is running in another task:

```

$SYBASE/$SYBASE_OCS/bin/isql -U sa -P
1> select @@spid
2> go
-----
12
1>

$SYBASE/$SYBASE_ASE/bin/sqldbgr -U sa -P -S MERCURY:16896

(sqldbg) attach 13
The spid is invalid
(sqldbg) attach 12
(sqldbg) show breakpoints
(sqldbg) stop in sp_who
Breakpoint moved to line 20
(sqldbg) /* at this point run the sp_who procedure from spid 12 */
(sqldbg) where
(sp_who::20::@loginame = <NULL>)
(ADHOC::1::null)
(sqldbg) next
(sp_who::22) set chained off
(sqldbg) next
(sp_who::25)set transaction isolation level 1
(sqldbg) cont
(sqldbg) /* at this point the sp_who result will show up in the isql screen */
(sqldbg) detach 12
(sqldbg)

```

Usage

- The sql command is executed in the context of debugged task, while the mysql command is executed in the context of debugger task. Setting session-specific information, such as for set quoted\_identifier on through sql does not work.
- By default, the Sybase jConnect JDBC driver uses set quoted\_identifier on. Since the sqldbg utility is built using jConnect arguments that need quotes, use single quotes instead of double quotes when entering options. For example, use sp\_configure 'allow update' instead of sp\_configure "allow update".
- Before you can run sqldbg, you must set either the SYBASE\_JRE or JAVA\_HOME environments to the location containing the Java run environment.
- When you invoke sqldbg at the command prompt, the utility starts and the prompt changes to a sqldbg prompt:

(sqldbg)

Once you see the (sqldbg) prompt, you can enter the following sqldbg commands to perform your tasks:

**Table 7-4: sqldbg commands and their descriptions**

Command	Description
attach <i>spid</i>	Attaches a task to sqldbg when you are already logged in to Adaptive Server.  <b>Note</b> Do not use attach <i>spid</i> to attach to a procedure that is not running.  sqldbg cannot debug multiple tasks in the same session. If you try to attach the utility to multiple tasks, the first <i>spid</i> continues to be marked as attached. Since you cannot attach to a <i>spid</i> that is already attached, you must use the detach command, and then attach to another <i>spid</i> .
run <i>procname</i>	Debugs stored procedures and triggers without attaching sqldbg to an existing task. If you attempt to use run <i>procname</i> while you are already debugging an existing task with attach <i>spid</i> , run <i>procname</i> fails and you see the following:  Cannot run a procedure while debugging another task
stop in <i>procname</i> [at line #]	Sets a breakpoint to stop the stored procedure or trigger being debugged at the beginning of the specified procedure name.  stop in <i>procname</i> at line # sets a breakpoint to stop the stored procedure or trigger being debugged at a designated line within the specified procedure.  If you enter an invalid line number, sqldbg moves the breakpoint to the next valid line number, and displays:  Invalid line number  You can also use this command to set multiple breakpoints.

Command	Description
show breakpoints	<p>Displays the breakpoint handle in the form of a unique number, as well as the breakpoint statements given by the user during the sqldbgr session.</p> <p>If you specify a breakpoint line number that does not contain a valid SQL statement, Adaptive Server moves the breakpoint to the next valid line number. However, Adaptive Server does not change the command you entered. This is why show breakpoints can return a breakpoint handle and a breakpoint statement given during the sqldbgr session that can be different.</p> <p>An asterisk (*) in the breakpoint line indicates that the breakpoint is set, but currently disabled.</p>
use <i>dbname</i>	Tells sqldbgr what database to use in order to debug that database's stored procedures or triggers.
show variables [at level #] show @ <i>varname</i> [at level #]	<p>show variables displays all the variables and their values in the current SQL stored procedure or trigger.</p> <p>show variables at level # displays the variables and their values in the current SQL stored procedure or trigger at the specified level.</p> <p>show @ <i>varname</i> displays the indicated variable and its value in the current SQL stored procedure or trigger.</p> <p>show @ <i>varname</i> at level # displays the indicated variable and its value in the current SQL stored procedure or trigger at the specified level.</p> <hr/> <p><b>Note</b> sqldbgr does not support Java variables.</p>
show where	Displays the call stack of the stored procedures and triggers that exist in the task being debugged.
step or next	step or next instructs sqldbgr to move to the next statement in the current stored procedure or trigger.
step into	Instructs sqldbgr to move into a procedure if the current statement is an execute statement. If the current statement is an update, delete, or insert statement, and if there are triggers in it, step into instructs sqldbgr to move into the update, delete, or insert triggers.
step out	Instructs sqldbgr to move out of the current stored procedure or trigger, and to stop at the next line in the calling procedure.
set @ <i>varname</i> = VALUE	Sets the value of the indicated variable to the variable value declared in the command in the current stored procedure or trigger. The values for the variables set using set @ <i>varname</i> = VALUE are valid only for the current session sqldbgr.
cont[inue]	Instructs sqldbgr to continue debugging, and to stop at the next breakpoint (if any).
delete #	Deletes the indicated breakpoint set in the current instance of sqldbgr.
enable # and disable #	Enables the indicated breakpoints. disable # does the opposite.
sql <i>any_sql_statement</i>	<p>Executes ad hoc SQL statements. You can use this command to select and analyze data from temp tables created by the task being debugged.</p> <p>sql <i>any_sql_statement</i> returns a result set and any errors that occurred.</p>

Command	Description
detach <i>spid</i>	Detaches sqldbg from the indicated <i>spid</i> , and releases the task being debugged. It deletes the breakpoints that were set for the task being debugged during the current sqldbg session.
help [all]	Display sqldbg commands.

Table 7-5 lists all sqldbg error messages:

**Table 7-5: sqldbg error messages and their meaning**

Error message	Description
Cannot allocate resource in ASE	Indicates that Adaptive Server does not have sufficient memory resources to execute sqldbg. Increase procedure cache size and restart sqldbg.
Cannot create Debugger handle in ASE	Indicates that Adaptive Server does not have sufficient memory resources to create a debugger handle. Increase procedure cache size and restart sqldbg.
The <i>spid</i> is invalid	Displays when you attempt to attach sqldbg to an invalid <i>spid</i> . Double check the <i>spid</i> and try again.
You cannot debug a task that is not owned by you	Displays when you try to debug a task that you do not own. You must log in to the server as the owner of the task to be debugged.
<i>Spid</i> is already being debugged	Displays when you execute attach <i>spid</i> and attempt to attach to a <i>spid</i> that is already being debugged.
<i>Spid</i> is not debugged currently	Displays when you execute detach <i>spid</i> and attempt to detach from a <i>spid</i> that is not attached to sqldbg.
Invalid command	Displays when you enter an invalid command.
Invalid procedure name	Displays when you enter an invalid procedure name in stop in <i>procname</i> .
Invalid line number	Displays when you enter an invalid line number in stop in <i>procname</i> at line #.
Variable not found	Displays when you enter an invalid variable in show @ <i>varname</i> , show @ <i>varname</i> at level #, or set @ <i>varname</i> = VALUE.
Illegal conversion attempted	Displays when you execute set @ <i>varname</i> = VALUE and attempt to convert the variable to an invalid value.
Conversion from text to datatype failed	Displays when set @ <i>varname</i> = VALUE is unsuccessful.
Cannot run a procedure while debugging another task	Displays if you use run <i>procname</i> while already debugging an existing task with attach <i>spid</i> .

## sqlloc

Description	<b>UNIX platforms only</b> Installs and modifies languages, character sets, and sort order defaults for Adaptive Server using a GUI based on X11/Motif. sqlloc is located in <code>\$\$SYBASE/\$SYBASE_ASE/bin</code> .
Syntax	<pre>sqlloc     [-S <i>server</i>]     [-U <i>user</i>]     [-P <i>password</i>]     [-s <i>sybase dir</i>]     [-I <i>interfaces file</i>]     [-r <i>resource file</i>]</pre> <p>Or</p> <pre>sqlloc -v</pre>
Parameters	<p><b>-S <i>server</i></b> specifies the name of the Adaptive Server to which to connect.</p> <p><b>-U <i>user</i></b> specifies a login name. Logins are case sensitive.</p> <p><b>-P <i>password</i></b> specifies the “sa” account password.</p> <p><b>-s <i>sybase dir</i></b> specifies the value to use for the SYBASE environment variable.</p> <p><b>-I <i>interfaces file</i></b> specifies the name and location of the interfaces file to search when connecting to Adaptive Server.</p> <p><b>-r <i>resource file</i></b> executes the specified resource file.</p> <p><b>-v</b> prints the version number and copyright message for sqlloc and then exits.</p>
Usage	<ul style="list-style-type: none"> <li>You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use sqlloc.</li> <li>You must set the DISPLAY environment variable before invoking sqlloc, unless you are only using the -v parameter to display the version number.</li> </ul>
Permissions	You must be a Sybase System Administrator to use sqlloc.
See also	<p><b>Documentation</b> <i>Installation Guide for UNIX Platforms</i> for more information about sqlloc.</p> <p><b>Utilities</b> langinstall, sqllocres</p>

## sqllocres

Description	<b>UNIX platforms only</b> Installs and modifies languages, character sets, and sort order defaults for Adaptive Server, using a resource file. sqllocres is located in <code>\$\$SYBASE/\$SYBASE_OCS/bin</code> .
Syntax	sqllocres [-S <i>server</i> ] [-U <i>user</i> ] [-P <i>password</i> ] [-s <i>sybase dir</i> ] [-l <i>interfaces file</i> ] [-r <i>resource file</i> ]  Or  sqllocres -v
Parameters	-S <i>server</i> specifies the name of the Adaptive Server to which to connect.  -U <i>user</i> specifies a login name.  -P <i>password</i> specifies the “sa” account password.  -s <i>sybase dir</i> specifies the value to use for the SYBASE environment variable.  -l <i>interfaces file</i> specifies the name and location of the interfaces file to search when connecting to Adaptive Server.  -r <i>resource file</i> executes the specified resource file.  -v prints the version number and copyright message for sqllocres, then exits.
Usage	You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use sqllocres.
Permissions	You must be a Sybase System Administrator to use the sqllocres utility.
See also	For more information about the sqllocres utility program, see the <i>Installation Guide for UNIX Platforms</i> .  <b>Utilities</b> langinstall, sqlloc

## sqlsrvr

Description	<b>Windows platforms only</b> The executable form of the Adaptive Server program, this utility is located in %SYBASE%\%SYBASE_ASE%\bin.
Syntax	<pre>sqlserver [-f] [-g] [-G] [-h] [-H] [-m] [-P] [-q] [-v] [-X]           [-a path_to_CAPs_directive_file]           [-b master_device_size] [k   K   m   M   g   G   t   T ]           [-c config_file_for_server]           [-d device_name]           [-e path_to_error_log]           [-i interfaces_file_directory]           [-K keytab_file]           [-L config_file_name_for_connectivity]           [-M shared_memory_repository_directory]           [-p sa_login_name]           [-r mirror_disk_name]           [-s server_name]           [-T trace_flag]           [-u sa/sso_name]           [-w master   model database]           [-y [password] ]           [-z page_size [ k   K ] ]</pre>
Parameters	<p><b>-f</b> forces initialization of a device or database. You must use both <b>-b</b> and <b>-w</b> to use <b>-f</b>.</p> <p><b>-g</b> turns off event-logging.</p> <p><b>-G</b> specifies the name of the event log server.</p> <p><b>-h</b> prints this help message, then exists.</p> <p><b>-H</b> starts the High Availability (HA) server, if you have the HA feature installed on your Adaptive Server.</p> <p><b>-m</b> starts Adaptive Server in single-user mode.</p> <p><b>-q</b> treats quiesced databases as “in recovery.”</p> <p><b>-v</b> prints the version number and copyright message for sqlsrvr and then exits.</p>

- X  
starts this server as sybmon, not dataserver.
- a *path\_to\_CAPs\_directive\_file*  
specifies the path to the CAPs directive file.
- b *master\_device\_size [k | K | m | M | g | G | t | T]*  
specifies the size of the master device.
- c *config\_file\_for\_server*  
specifies the full path name of an Adaptive Server configuration file. Use this parameter to start Adaptive Server with the configuration values in the specified configuration file.

If you specify a configuration file with the sqlsrv -c parameter, make sure all the parameters in this configuration file are compatible before you boot the server. If some of the configuration parameters are incompatible, the server may not boot. To avoid this, do not specify a configuration file when you build the master device. The build phase uses all default settings when you do not specify a configuration file.

For more information, see the *System Administration Guide*.

- d *device\_name*  
is the full path name of the device for the master database. The master database device must be writable by the user who starts Adaptive Server. The default master database device name is d\_master.
- e *errorlogfile*  
is the full path name of the error log file for Adaptive Server system-level error messages.
- i *interfaces\_file\_directory*  
specifies the directory location of the interfaces file to search when connecting Adaptive Server. If -i is omitted, sqlsrv looks for a file named *interfaces* in the directory pointed to by your SYBASE environment variable.
- K *keytab\_file*  
specifies the path to the keytab file used for authentication in DCE.
- L *config\_file\_name\_for\_connectivity*  
specifies the name the configuration file for connectivity.



- M *sharedmem\_directory*  
places shared memory files in the specified directory instead of in the default location, *%SYBASE%*. If *sharedmem\_directory* starts with “\”, the directory name is assumed to be absolute. Otherwise, the directory name is interpreted relative to *%SYBASE%*.
- p *sso\_login\_name*  
specifies the login name of a System Security Officer when starting Adaptive Server, for the purposes of getting a new password for that account. Adaptive Server generates a random password, displays it, encrypts it, and saves it in *master..syslogins* as that account’s new password.
- r *mastermirror*  
starts the mirror of the master device. Use this parameter to start Adaptive Server if the master device has been damaged.
- s *servername*  
specifies the name of the Adaptive Server to start. If -s is omitted, a server named SYBASE is started.
- T *trace\_flag*
- u *sa/sso\_name*  
specifies the System Administrator or System Security Officer’s name you want to unlock.
- w *master | model\_database*  
specifies whether you want to write a master or model database.
- y [*password*]  
allows you to assign a password for the encrypted private key, so that the server prompt the user for a password. This password should match the password you used to encrypt the private key when it was created. You cannot use this parameter when you are running the server in the background.

---

**Note** Although you can a password with -y, for security reasons Sybase strongly discourages you from doing so.

---

A private key is included with your server’s digital certificate. By default, the certificate file located:

```
%SYBASE%\%SYBASE_ASE%\certificates\servername.crt
```

The location of the certificate file changes if you invoke the `sp_ssladmin addcert` command.

`-z page_size`

specifies the page size of the server. You must use `-b` and `-w` to use this flag, and name an even power of two between 2k and 16k, or else the server does not boot.

## Examples

**Example 1** Creates a new installation with a 100MB master device and a 4k page:

```
sqlsrvr -d d_master -z 4k -b 100.02M
```

The spaces between options and their following arguments are optional and acceptable. This example specifies “100.02M” for a 100MB master device because the server requires 16KB of overhead for its configuration area.

**Example 2** Rewrites a corrupt model database:

```
sqlsrvr -d d_master -w model
```

**Example 3** Rewrites a corrupt master database, specifying device size:

```
sqlsrvr -d d_master -w master -z 4k
```

**Example 4** Rewrites a corrupt master database, specifying device and page sizes, forcing the server to accept these values in preference to what it may find in the config block:

```
sqlsrvr -d d_master -w master -z 4k -b 100.02M -f
```

**Example 5** Rewrites a corrupt master database, specifying a page size that does not match what the server finds in its config block. This produces a failure:

```
sqlsrvr -d d_master -w master -z 8k
```

```
00:00000:00000:2001/01/19 12:01:26.94 server The
configured server page size does not match that
specified on the command line. To use the configured
size, omit the command line size; to use the command
line size, specify 'force' (-f).
```

**Example 6** Rewrites a corrupt master database, specifying an incorrect page size, even in a normal boot. This produces a failure:

```
sqlsrvr -d d_master -z4000
```

```
sqlsrvr: the 'z' flag may not be used without 'b' or
'w'. sqlsrvr: server will ignore the 'z' flag. sqlsrvr:
```

the 'z' flag contained an invalid page size. `sqlsrvr`: the page size must be an even power of two between 2048 and 16384 bytes, inclusive.

#### Usage

- The `sqlsrvr` utility is referred to as `dataserver` in other Sybase documents.
- Start Adaptive Server using the services manager utility rather than by executing the `sqlsrvr` program directly. If you need to change any of the default parameters, edit the Adaptive Server's Registry keys. See the configuration guide for your platform for details.
- Adaptive Server derives its running environment from values in the `sysconfigures` system table. Run `sp_configure` to see the configuration values; use `sp_configure` and `reconfigure` to change the configuration.
- Because Adaptive Server passwords are encrypted, you cannot recover forgotten passwords. If all System Security Officers lose their passwords, the `-p` parameter generates a new password for a System Security Officer's account. Start Adaptive Server with `-p`, immediately log in to Adaptive Server with the new random password, and execute `sp_password` to reset your password to a more secure one.
- By default, Adaptive Server logs error messages in both the local error log file and the local Windows event log. You can disable Windows event logging by including the `-g` parameter and specifying a different event-logging machine with `-G machine_name`. Use standard Windows conventions when entering the `machine_name`. For example, to designate a PC named "LOGSITE", substitute "\\LOGSITE" for the `machine_name`. See the configuration guide for your platform for details on logging error messages.
- After you have finished running the installer, set the file permissions on the `sqlsrvr` executable to limit who can execute it.
- If you do not specify an Adaptive Server name with the `-s` parameter, and you have not set the `DSLISTEN` environment variable, `sqlsrvr` uses the default Adaptive Server name `SYBASE`. The value of the `DSLISTEN` environment variable overrides this default value, and the `-s` parameter overrides both the default and the `DSLISTEN` environment variable.
- Automatic login lockouts can cause a site to end up in a situation in which all accounts capable of unlocking logins (System Administrators and System Security Officers) are locked. If this occurs, use the `sqlsrvr` utility with the `-u` parameter to check the specified login for System Administrator or System Security Officer authorization, unlock the account, and reset the value of the current failed logins counter to zero.

- `-f` is only valid when used with `-b` and/or `-w`. The server fails to boot if you use `-f` without either `-b` or `-w`. `-f` forces the server in different ways, depending whether `-w` is present. See `-b` and `-w` below.

#### Starting Adaptive Server

Use either of the following methods to start Adaptive Server with a specified configuration file:

- Use Server Config to configure the server to have the `-c` parameter. In the Configure Adaptive Server window, select the Command Line option, and in the Command Line Parameters window, enter:

`-Cconfiguration_file_pathname`

For example, entering “`-chaze.cfg`” starts the server using the `haze.cfg` configuration file.

- Start Adaptive Server from the command line and provide the `-c` parameter.

#### Dependencies and conditions with `-b` and `-w`

The effect of `-b` changes depending on whether `-w` is present:

- `-b` without `-w` creates a new master device as named by `-d` (the default is `d_master`) and with the page size as specified by `-z` (the default is 2048):
  - If the named device already exists as an OS file, the attempt fails, and you must remove the existing file and try again.
  - If the named device names an existing raw partition, the attempt fails unless you include the `-f` flag. This reinitializes the raw partition as a server master device.
- `-b` with `-w` master tells `dataserver` to use the size specified in `-z` for the master device when recreating the master database. It implies nothing about creating a new device.

`-w` may or may not require additional flags:

- If you use `-w` `model`, the `-z` and `-b` flags are accepted but ignored.
- If you use `-w` `master` for **new** installations, `-z` and `-b` are not required because the device size information is stored in the `config_block`.
- If you use `-w` `master` to **upgrade** older installations:
  - The server requires `-b` and/or `-z` if the `config_block` does not contain a valid entry for the associated size(s). The command fails if it can't get valid data for the page size or device size.

- You may provide `-b` and/or `-z` when the `config_block` contains valid entries for the size(s) they represent. However if the sizes do not match what is in the `config_block`, you must add `-f` to force your new size preferences.
- `-f` may appear without either `-b` or `-z`, because `-f` also instructs the server to accept damaged allocation pages as belonging to the master database. This is useful for restoring badly corrupted databases. If you specify `-w master -f`, the server assigns to the master database every allocation page on the named master device that does not belong to some other database than master.

Permissions	Anyone with execute permission on the binary, and who has read/write access to all the files.
Tables used	sysconfigures
See also	<b>Commands</b> disk mirror, disk remirror, reconfigure <b>System procedures</b> sp_configure, sp_password <b>Utilities</b> startserver

## sqlupgrade

Description	<b>UNIX platforms only</b> Upgrades your currently installed version of Adaptive Server to the newest release using a GUI based on X11/Motif. sqlupgrade is located in <code>\$\$SYBASE/\$SYBASE_ASE/bin</code> .
Syntax	sqlupgrade [-s <i>sybase dir</i> ] [-r <i>resource file</i> ]  Or  sqlupgrade -v
Parameters	-s <i>sybase dir</i> specifies the value to use for the SYBASE environment variable.  -r <i>resource file</i> executes the specified resource file.  -v prints the version number and copyright message for sqlupgrade and then exits.
Usage	<ul style="list-style-type: none"><li>• You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use sqlupgrade.</li><li>• You must set the DISPLAY environment variable before invoking sqlupgrade, unless you are only using the -v parameter to display the version number.</li></ul>
Permissions	You must be a Sybase System Administrator to use sqlupgrade.
See also	<b>Documentation</b> <i>Installation Guide for UNIX Platforms</i> for more information about sqlupgrade.  <b>Utilities</b> preupgrade, sqlupgraderes

## sqlupgraderes

Description	<b>UNIX platforms only</b> Upgrades your currently installed release of Adaptive Server to the newest release using resource files. sqlupgraderes is located in <code>\$SYBASE/\$SYBASE_OCS/bin</code> .
Syntax	sqlupgraderes [-s <i>sybase dir</i> ] [-r <i>resource file</i> ]  Or  sqlupgraderes -v
Parameters	-s <i>Sybase Dir</i> specifies the value to use for the SYBASE environment variable.  -r <i>Resource File</i> executes the specified resource file.  -v prints the version number and copyright message for sqlupgraderes and then exits.
Usage	You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use sqlupgraderes.
Permissions	You must be a Sybase System Administrator to use sqlupgraderes.
See also	<b>Documentation</b> <i>Installation Guide for UNIX Platforms</i> for more information about sqlupgraderes.  <b>Utilities</b> sqlupgrade

## srvbuild

Description	<p><b>UNIX platforms only</b> Creates a new Adaptive Server, Backup Server, Monitor Server, or XP Server with default or user-specified values for key configuration attributes. You can use <code>srvbuild</code> in either GUI mode based on X11/Motif, or non-GUI mode. <code>srvbuild</code> is located in <code>\$\$SYBASE/\$\$SYBASE_ASE/bin</code>.</p>
Syntax	<pre> srvbuild     [-s <i>sybase_dir</i>]     [-l <i>interfaces_file</i>]     [-r <i>resource_file</i>] </pre> <p>Or</p> <pre> srvbuild -v </pre>
Parameters	<p><code>-s <i>sybase_dir</i></code>  specifies the value to use for the SYBASE environment variable.</p> <p><code>-l <i>interfaces_file</i></code>  specifies the name and location of the interfaces file to search when connecting to Adaptive Server.</p> <p><code>-r <i>resource_file</i></code>  executes the specified resource file.</p> <p><code>-v</code>  prints the version number and copyright message for <code>srvbuild</code> and then exits.</p>
Usage	<p>You must set the SYBASE environment variable:</p> <ul style="list-style-type: none"> <li>• To the location of the current version of Adaptive Server before you can use <code>srvbuild</code>.</li> <li>• Before invoking <code>srvbuild</code>, unless you are only using the <code>-v</code> parameter to display the version number.</li> </ul> <p>Using LDAP with <code>srvbuild</code> in a 64-bit environment</p> <p>When you use <code>srvbuild</code> to build a new server using a Lightweight Directory Access Protocol (LDAP) service in a 64-bit environment, you must edit the LDAP server entry.</p> <p><code>srvbuild</code> is a 32-bit application and uses the LDAP server entry from the <code>\$\$SYBASE/\$\$SYBASE_OCS/config/libtcl.cfg</code> file. Adaptive Server is a 64-bit application and uses the LDAP server information from the <code>\$\$SYBASE/\$\$SYBASE_OCS/config/libtcl64.cfg</code> file.</p>



Do not include any blank spaces after the LDAP server entry in the *libtcl.cfg* or *libtcl64.cfg* files; these prevent `srvbuild` from connecting to the LDAP server.

**Permissions**

You must be a Sybase System Administrator to use `srvbuild`.

**See also**

**Documentation** *Installation Guide for UNIX Platforms* for more information about `srvbuild`, and Chapter 11, “Managing Adaptive Server Logins, Database Users, and Client Connections,” in the *System Administration Guide* for more information on configuring a server using LDAP.

**Utilities** `srvbuildres`

## srvbuildres

Description	<b>UNIX platforms only</b> Creates, using resource files, a new Adaptive Server, Backup Server, Monitor Server, or XP Server with default or user-specified values for key configuration attributes. <code>srvbuildres</code> is located in <code>\$\$SYBASE/\$\$SYBASE_ASE/bin</code> .
Syntax	<pre> srvbuildres     [-s <i>sybase_dir</i>]     [-l <i>interfaces_file</i>]     [-r <i>resource_file</i>] </pre> <p>Or</p> <pre> srvbuildres -v </pre>
Parameters	<p><code>-s <i>sybase_dir</i></code>  specifies the value to use for the SYBASE environment variable.</p> <p><code>-l <i>interfaces_file</i></code>  specifies the name and location of the interfaces file to search when connecting to Adaptive Server.</p> <p><code>-r <i>resource_file</i></code>  executes the specified resource file.</p> <p><code>-v</code>  prints the version number and copyright message for <code>srvbuildres</code> and then exits.</p>
Usage	You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use <code>srvbuildres</code> .
Permissions	You must be a Sybase System Administrator to use <code>srvbuildres</code> .
See also	<b>Documentation</b> <i>Installation Guide for UNIX Platforms</i> for more information about <code>srvbuildres</code> .
	<b>Utilities</b> <code>srvbuild</code>

## startserver

Description	<b>UNIX platforms only</b> Starts an Adaptive Server or a Backup Server. startserver is located in <code>\$\$SYBASE/\$SYBASE_ASE/bin</code> .
Syntax	startserver <code>[[ -f runserverfile] [-m]] ...</code>
Parameters	<p><code>-f runserverfile</code>  specifies the relative path name of a runserver file, which is used as a reference each time you start an Adaptive Server or Backup Server. By default, the runserver file is in the current directory and is named <code>RUN_servername</code>. If you start a second Adaptive Server on the same machine, startserver creates a new runserver file named <code>RUN_servername</code>.</p> <p><code>-m</code>  starts Adaptive Server in single-user mode, allowing only one System Administrator to log in, and turns the allow updates to system tables configuration parameter on. Use this mode to restore the master database. The System Administrator can use the <code>dbo use only</code> parameter of <code>sp_dboption</code> for system administration activities that require more than one process, such as bulk copying or using the data dictionary. startserver normally starts up only one server per node.</p> <p>The <code>-m</code> parameter creates an <code>m_RUNSERVER</code> file and overwrites any existing <code>m_RUNSERVER</code> file.</p>
Examples	<p><b>Example 1</b> Starts an Adaptive Server named SYBASE from the runserver file named <code>RUN_servername</code> in the current directory:</p> <pre>startserver</pre> <p><b>Example 2</b> Starts an Adaptive Server named MYSERVER and a Backup Server named SYB_BACKUP:</p> <pre>startserver -f RUN_MYSERVER -f RUN_SYB_BACKUP</pre> <p><b>Example 3</b> Starts only the Backup Server SYB_BACKUP:</p> <pre>startserver -f RUN_SYB_BACKUP</pre>
Usage	<ul style="list-style-type: none"> <li>startserver uses the information in the runserver file to start an Adaptive Server or Backup Server. The master device must be writable by the user who starts Adaptive Server.</li> </ul>

The `startserver` command creates the Adaptive Server error log file (named *errorlog*) in the directory where the server is started, and adds this information as part of the `-e` parameter in the Adaptive Server executable line in the `runserver` file. If a second Adaptive Server is started on the same machine, a new error log named *errorlog\_servername* is created; this information is added to that server's `runserver` file. The user must have execute permission on the specified `runserver` file.

- You can start multiple servers by specifying more than one `runserver` file, as shown in example 2. You can specify `-m` after each `-f runserverfile`.
- Adaptive Server derives its running environment from values in the `config` file. Run `sp_configure` or edit the `config` file to see or change configuration parameters.
- To ensure the integrity of your Adaptive Server, it is important that you apply appropriate operating-system protections to the `startserver` executable and the `runserver` file.

#### The `runserver` file

- The `runserver` file, which is created by `srvbuild` during installation, contains the `dataserver` command to start Adaptive Server or the `backupserver` command to start Backup Server. By default, the `runserver` file is in the current directory and is named *RUN\_servername*. You can edit the `runserver` file to correct the options and parameters for the commands. The following example shows two sample `runserver` files.

#### Runserver file for server MYSERVER:

```
#!/bin/sh
#
# Adaptive Server Information:
# name: /MYSERVER
# master device: /remote/Masters/myserver_dat
# master device size: 10752
# errorlog: /remote/serverdev/install/errorlog
# interfaces: /remote/serverdev/interfaces
#
#
/$SYBASE/$SYBASE_ASE/bin/dataserver -d/remote/Masters/myserver_dat \
-sMYSERVER -e/remote/serverdev/install/MYSERVER_errorlog \
-i/remote/serverdev &
```

#### Runserver file for backup server SYB\_BACKUP:

```
#!/bin/sh
#
# Backup Server Information:
```

```
# name: SYB_BACKUP
# errorlog: /remote/serverdev/install/backup.log
# interfaces: /remote/serverdev/interfaces
# location of multibuf: /remote/serverdev/bin/sybmultbuf
# language: us_english
# character set: iso_1
# tape configuration file: /remote/serverdev/backup_tape.cfg
#
#
/remote/serverdev/bin/backupserver -SSYB_BACKUP \
-e/remote/serverdev/install/backup.log \
-I/remote/serverdev/interfaces \
-M/remote/serverdev/bin/sybmultbuf -Lus_english -Jiso_1 \
-c/remote/serverdev/backup_tape.cfg
```

See also **Commands** disk mirror, disk remirror, disk unmirror  
**Utilities** backupserver, dataserver

## sybmigrate

Description	<p>sybmigrate allows you to convert an Adaptive Server from one page size to another page size, and to migrate between platforms. The executable file is located in the following directory:</p> <p><b>UNIX platforms</b> <code>\$\$SYBASE/\$SYBASE_ASE/bin/sybmigrate</code></p> <p><b>Windows</b> <code>%%SYBASE%\%SYBASE_ASE%\bin\sybmigrate.bat</code></p>
Syntax	<pre>sybmigrate [-v ] [-h ] [-f ]             [-D 1   2   3   4 ]             [-l <i>interfaces_file</i> ]             [-r <i>input_resource_file</i> ]             [-m setup   migrate   validate   report ]             [-rn status   space_est   repl   diff   password ]             [-l <i>log_file</i> ]             [-t <i>output_template_resource_file</i> ]             [-J <i>client_charset</i> ]             [-z <i>language</i> ]             [-T <i>trace_flags</i> ]             [-Tase <i>trace_flags</i> ]             [-f ]</pre>
Parameters	<p>-v prints the version string and exits.</p> <p>-h prints the help information and syntax usage and exits.</p> <p>-f overrides the locking session.</p> <p>-D sets the debug level for sybmigrate. The default debug level is 2.</p> <p>-l identifies a specific <i>interfaces</i> file to find server names. If no <i>interfaces</i> file location is designated, sybmigrate uses:</p> <ul style="list-style-type: none"><li>• For UNIX – <code>\$\$SYBASE/interfaces</code></li><li>• For Windows – <code>%%SYBASE%\ini\sql.ini</code></li></ul> <p>-r specifies that the resource file mode is to be used in the migration process. If the input resource file is not specified by using the -r parameter, sybmigrate operates in GUI mode.</p>

-m

designates the types of operations that are performed:

- **setup** – to set up the repository and migration working database, and to migrate the server-wide data.
- **migrate** – to perform data and object migration.
- **validate** – to validate the migrated objects.
- **report** – to run any of the five reports. The reports can be run in the GUI and resource file mode. The available reports are:
  - **status** – the migrate object status report gives information about objects that have been migrated.
  - **space\_est** – use the target database space estimation report to verify that you have sufficient resources allocated to your target database.
  - **repl** – use the replication report to check any explicitly replicated objects that have been migrated, determine the type of replication system, and to produce SQL commands for users to execute on the target Adaptive Server and the Replication Server.
  - **diff** – checks the objects between the source and target databases. The diff report provides the following information for the following object types:
    - Server information
    - Database information
    - DDL objects
    - User table data
- **password** – creates a file for the changed passwords.

-rn

indicates what type of report to generate. If -rn is not specified, all five reports are run.

-l

indicates a user-defined log file where the output of the migration process is stored. If -l is not used, the logs are stored in `$$SYBASE/$SYBASE_ASE/init/logs` or the working directory.

- t  
directs sybmigrate to generate an output template resource file, to be used for subsequent migrations in the resource file mode.
- J  
specifies the character set to be used for the Adaptive Server connection.
- z  
specifies the language to be used for the Adaptive Server connection.
- T  
sets command line trace flags.
- Tase  
is used to run Adaptive Server trace flags (turned on using dbcc traceon) for all Adaptive Server connections opened by sybmigrate. The trace flags should be specified in a comma-separated list.

**Examples**

**Example 1** Runs the status report:

```
sybmigrate -r resource file -m report -rn status
```

**Example 2** Runs the space\_est report in the resource file mode:

```
sybmigrate -r resource file -m report -rn space_est
```

**Example 3** Runs the repl report, issue:

```
sybmigrate -r resource file -m report -rn repl
```

**Usage**

- Make sure the allow resource limits configuration parameter is set to 0 before running sybmigrate.
- You cannot migrate server data if metadata already exists on the target Adaptive Server.
- If sybmigrate exited a session inappropriately, use -f to override the source and target database binding that is created so that only one session of sybmigrate can run on a source and target database path.
- If you use the -r parameter, then you also need to use the -m argument to specify the type of operation to perform: setup, migrate, validate, or report. You can run the entire migration process in the resource file mode, or you can choose to run only parts of in this fashion.
- You can use -t only in the resource file mode.  
  
-t requires that you start sybmigrate using the -r argument specifying the login information. This argument also requires -m to specify what type of resource file is to be generated.



- Adaptive Server versions 12.5.3 and later allow you to specify the size and location of a work database on your target server. When migrating a database or server from a source server with Adaptive Server Enterprise versions 12.0 and later but earlier than 12.5.0.1, you must specify the size and location of a work database on the target server.
- You can override `sybmigrate`, and use the `interfaces` file by providing the `-l` argument if the LDAP entry is defined in:
  - UNIX – `$$SYBASE/$SYBASE_OCS/config/libtcl.cfg`
  - Windows – `%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg`

### Reports

- `status` – the migrate object status report gives information about objects that have been migrated.
- `space_est` – use the target database space estimation report to verify that you have sufficient resources allocated to your target database.
- `repl` – use the replication report to check any explicitly replicated objects that have been migrated, determine the type of replication system, and to produce SQL commands for users to execute on the target Adaptive Server and the Replication Server.
- `diff` – checks the objects between the source and target databases. Users can run the report on individual objects, or the entire database, except for server and database information or metadata. You can run the `diff` report at any time. You do not need to run a setup session to run the `diff` report. The source and target database name do not need to be the same when running the `diff` report.

The `diff` report provides the following information for the following object types:

- Server information – compares the master database system catalogs row count between the source and target Adaptive Server. This task is similar to the validation session.
- Database information – compares the user database system catalogs row count between the source and target Adaptive Server. This task is similar to the validation session.
- DDL objects – the report displays whether the objects exist on the source or the target Adaptive Servers. If the objects exists in both databases, that object is not displayed in the report.

- User table data – compares the row count of the user tables in the source and target Adaptive Server. If the table only exists in the source or target databases, the table is not displayed in the report.

Permissions

You must be a Sybase System Administrator or log in with the sa\_role to use sybmigrate.

See also

**Documentation** Chapter 8, “Migration Utility” for detailed information on sybmigrate.

## xpserver

Description	Starts XP Server manually.
Syntax	<pre>xpserver -S XP_Server xpserver -SXP_Server   [-linterfaces_file]   [-ppriority]   [-sstack_size]   [-u]   [-v]   [-x]</pre>
Parameters	<p><b>-S <i>XP_Server</i></b>  specifies the name of the XP Server to start. The format of the XP server name is <i>SQLSERVERNAME_XP</i>, where <i>SQLSERVERNAME</i> is the name of the Adaptive Server to which the XP Server is dedicated. For example, the XP Server for an Adaptive Server named SMOKE would be named SMOKE_XP. The XP Server name must be in uppercase.</p> <p><b>-l <i>interfaces_file</i></b>  specifies the name and location of the directory containing the interfaces file (<i>sql.ini</i>) that Adaptive Server searches when connecting to XP Server. If you do not specify -l, xpserver uses the <i>ini</i> subdirectory of the <i>%SYBASE%</i> release directory.</p> <p><b>-p <i>priority</i></b>  specifies the priority of the Open Server process. Values between 0 (lowest) and 15 (highest) are valid. Overrides the esp execution priority configuration parameter. The default is 8.</p> <p><b>-s <i>stack_size</i></b>  specifies (in bytes) the stack size of the process used to execute an extended stored procedure (ESP). Overrides the esp execution stacksize configuration parameter if it is set. The default is 34816 bytes.</p> <p><b>-u</b>  specifies that the functions be automatically unloaded from XP Server memory after the ESP request terminates. Overrides the esp unload dll configuration parameter if it is set. The default is not to unload the function.</p> <p><b>-v</b>  prints the version number and copyright message for XP Server and then exits.</p>

	<p>-x</p> <p>specifies that the client security context be used to execute operating system commands using the system ESP, xp_cmdshell. Overrides the xp_cmdshell context configuration parameter if it is set. The default is to use the security context of the operating system account of the Adaptive Server session.</p>
Usage	<ul style="list-style-type: none"><li>• XP Server is normally started automatically by Adaptive Server. Use the manual command to start XP Server <b>only</b> when instructed to do so in an “XP Server Failed to Start” error message.</li><li>• There can be only one XP Server per Adaptive Server. An Adaptive Server running ESPs communicates with a single XP Server, and the ESPs execute synchronously.</li><li>• The -p parameter affects the priority used by the Open Server scheduler. If -p is set to a high number, the scheduler can run XP Server before running the other threads <i>in</i> its run queue. If -p is set to a low number, the scheduler can run XP Server only when there are no other Open Server threads in its run queue. This parameter is unrelated to the application queue priorities within Adaptive Server, which are set by sp_bindexclass.</li></ul> <p>See the discussion of multithread programming in the <i>Open Server Server Library/C Reference Manual</i> for information about scheduling Open Server threads.</p> <ul style="list-style-type: none"><li>• If automatic unloading of ESP functions is not set by the -u parameter or by the esp unload dll configuration parameter, you can unload them at runtime using sp_freedll.</li><li>• Unlike Adaptive Server and Backup Server, XP Server does not have a runserver file.</li><li>• When configuring an XP Server, the directory service entry name must end with “_XP” in upper case, such as “abcdef_XP” or “ABCDEF_XP.”</li></ul>
Permissions	No special permissions are required to run xpserver.
See also	<b>System ESP</b> xp_cmdshell <b>System procedures</b> sp_configure, sp_freedll

This chapter discusses sybmigrate.

Topic	Page
Overview	259
Before you begin	263
Migration process	266
Sybmigrate with Encrypted Columns	291
Post-migration activities	292
Migrating databases in the Replication Server domain	293
Limitations	300
Troubleshooting and error messages	302

## Overview

sybmigrate is the migration tool used to migrate data from one server to another.

By default, sybmigrate migrates encrypted columns in ciphertext format. This avoids the overhead of decrypting data at the source and encrypting at the target. In some cases, sybmigrate chooses the reencrypt method of migration, decrypting data at the source and encrypting at the target.

---

**Note** When migrating from one Adaptive Server to another more recent version of Adaptive Server, you must specify the size and location of a work database on the target server.

---

## Benefits of *sybmigrate*

sybmigrate:

- Aids users in changing the page sizes of their database applications.

- Provides a manageable and smooth migration process.
- Allows customers to take advantage of the variable page size feature for existing databases with user data, thus realizing the full benefit of Adaptive Server versions 12.5 and later.

## What *sybmigrate* does

During the setup portion of the migration process, the following server data is migrated to the target Adaptive Server:

- Remote servers
- Logins
- Login attributes
- Server roles
- Login roles
- Role attributes
- Users
- Alternate users
- Roles
- Permissions
- Remote logins
- External login attributes
- Timer
- Resource limits
- Replication attributes
- Display level attributes
- User messages in the master database
- Java classes in the master database
- JAR files in the master database
- Proxy objects

During the migration portion of the migration process, the following database-specific data is migrated to the target database:

- Defaults
- User-defined datatypes
- Rules
- User tables
- User table data
- Views
- Triggers
- Indexes
- Stored procedures
- Extended stored procedures
- Users
- Logins
- Roles
- Remote servers
- Database data
  - Users
  - Alternate users
  - Roles
  - Role attributes
  - Permissions
  - User messages
  - Java classes
  - JAR files
- Defaults
- Rules
- User-defined types
- Tables

- Indexes
- Referential constraints
- Views
- Stored procedures
- Triggers

## What *sybmigrate* does not do

The following items must be migrated manually:

- Table-level lock promotion attributes
- User-defined thresholds
- Abstract plan definitions maintained in sysqueryplans
- All system databases except the model database
- Any required database options like cache binding, recovery order, and the associated log I/O size as specified by sp\_logiosize
- Proxy databases
- Engine groups
- Engine bindings
- Execution classes
- Cache configurations
- Auditing tables and auditing configuration
- Server-wide row-lock promotion settings
- Access rules

---

**Note** Drop access rules before beginning data migration; they can prevent the Database Owner from accessing all rows in a table, which prevents complete data migration.

---

- Compiled objects with hidden SQL text
- User-defined segments



- Constraints **are** migrated but when they are bound by name to user-defined message numbers, the bindings must be re-created manually
- Settings for objects such as ascinserts, indextrips, oamtrips, datatrips, and sortbufsize created using dbcc tune
- Device definitions
- SQLJ functions
- Proxy tables for external files
- Audit options and audit events
- Server configuration
- Database suspect threshold
- Recovery orders

## Before you begin

### Required components for the *sybmigrate*

*sybmigrate* requires JRE 1.4, jConnect™ for JDBC™ 6.0, ddlgen components, and Component Integration Services in the source Adaptive Server.

Because *sybmigrate* requires a server-to-server connection, two Adaptive Servers must be running. Make sure that you have the appropriate licenses.

### Dependencies

Before you begin the migration process, create databases, devices, and segments on the target Adaptive Server. Server and cache configurations must also be already installed on the target Adaptive Server.

Use *ddlgen* to extract the corresponding scripts from the source Adaptive Server, and modify them as needed before applying them to the target Adaptive Server. For more information, see *ddlgen* on page 163.

## Installation

sybmigrate is installed as part of the Adaptive Server software. For information about how to install Adaptive Server, see the *Installation Guide* for your platform.

## Permissions

The System Administrator login is needed for the setup portion of the migration process. For the remainder of the process, the login must have “sa\_role” and “sso\_role” privileges to run sybmigrate.

## Changing target login accounts

Once you have migrated between different platforms, login passwords are not compatible. However, sybmigrate allows you to change the password on target Adaptive Server login accounts during the setup session of the migration process in either of two ways:

- Let sybmigrate generate a password on the target server. sybmigrate outputs a list of passwords used during migration after the process is complete.
- Supply a password file that contains user name and password pairs. sybmigrate sets these passwords on the target server. Thereafter, the System Administrator must run sp\_password to create a new password for each login not included in the password file.

---

**Note** After the migration process is complete, the System Administrator must change passwords manually on the target Adaptive Server. The System Administrator must issue sp\_password for new login and for each login not reset during the migration process.

---

In addition to the changing password options, sybmigrate also allows you to lock and unlock target Adaptive Server accounts. This option is provided so that the System Administrator can block a user from logging into the target Adaptive Server during the migration process.

## Platforms

sybmigrate works on both UNIX and Windows platforms.

- For UNIX, the executable file is located in `$SYBASE/$SYBASE_ASE/bin/sybmigrate`.
- For Windows, the executable file is located in `%SYBASE%\%SYBASE_ASE%\bin\sybmigrate.bat`.

## Environment settings

The following environment variables must be set correctly. With the exception of SYBMIGRATE\_MEMORY, these environment variables are defined in the `SYBASE.csh` or `SYBASE.sh` files that are created during the installation process.

- SYBASE – defines the location of the Sybase release path.
- SYBASE\_ASE – defines the location of the Adaptive Server component directory.
- SYBASE\_JRE – defines the location of the Java runtime environment. This is generally set to `$SYBASE/shared/jre-1_4` in the Adaptive Server release area. This environment variable overrides JAVA\_HOME.

SYBASE\_JRE defaults to `$SYBASE/shared/jre142` (UNIX) and `%SYBASE%\Shared\Sun\jre142` (Windows).

- SYBMIGRATE\_MEMORY – specifies the amount of memory to be used when invoking the Java virtual machine (JVM). This environment variable should be specified with a number, which refers to the amount of memory in megabytes. If SYBMIGRATE\_MEMORY is not set, JVM uses the default memory setting of 512MB.

If sybmigrate is using a large number of threads, or working on many tables or indexes in parallel, increase the amount of memory allocated to the JVM on the client side.

## Migrating proxy tables

sybmigrate supports the migration of proxy tables. If you are planning to migrate proxy tables, you should do the following before you begin migration.

- Make sure that the remote servers involved in proxy table definitions is present in the target server interface file.

- In order to verify the DDL execution time, the remote server has to be accessible when the migration is performed.

## Migration process

The goal of `sybmigrate` is to provide a means to migrate all objects and user data that exist on the source Adaptive Server. However, when migration takes place, there is some server-wide data that needs to be migrated before any user data or user objects can be migrated to individual databases.

The hierarchy of objects dictates the order in which objects are re-created. Generally, server-wide objects from the master database are created first. Independent objects like default languages and character into databases first.

## Overview of the migration process

The migration procedure consists of configuring the source and target Adaptive Servers, setting up the migration paths, migrating objects, and validating the migrated objects.

The setup session establishes the migration paths from the source database to the target database. The setup creates the repository database and the work databases, and registers the option to migrate the server data. The setup session can only be executed by an “sa” login.

The migrate session is used to migrate objects and data from the source database to the target database.

The validate session validates the migrated objects. Validation ensures the integrity of data and objects that have been successfully migrated from the source database to the target database.

`sybmigrate` does not migrate an archive database if an entire installation is being migrated.

`sybmigrate` migrates an archive database only if the archive database is specifically selected for migration. When you migrate an archive database to a target server, `sybmigrate` automatically creates a traditional database—rather than an archive database—on the target server.

## Pre-migration considerations

You must have the source Adaptive Server and the target Adaptive Server running concurrently when you migrate data from one to the other.

`sybmigrate` assumes that the target Adaptive Server has been installed and configured prior to data migration. Use `srvbuild` or `syconfig` to create a new Adaptive Server with the required logical page size.

Keep the following items in mind prior to migration, when you are creating the target Adaptive Server and configuring the source Adaptive Server:

- `sybmigrate` requires allow resource limits to be set to 0.
- If metadata already exists on the target server, you cannot migrate server data.
- When you create a new Adaptive Server with a different logical page size into which you want to migrate data, you must adequately adjust the size of the database on the target Adaptive Server to accommodate the inbound data. If you are migrating data to an Adaptive Server with a larger logical page size, this is especially important.

Use the space estimation report, `space_est`, to determine how much space is available on your target database. For more information about `space_est`, see “Starting `sybmigrate`” on page 273.

- To speed the migration process, you can run multiple sessions of `sybmigrate` within the same server. However, running more than one session of `sybmigrate` on the same source and target database path is not allowed.
- You must manually create segments on the target database before migrating tables and indexes.
- The data transfer rate for `sybmigrate` is configured through CIS bulk insert array size. The default configuration for CIS bulk insert array size is 50 rows. This means that as many as 50 rows of data are buffered by CIS before being transferred to the target Adaptive Server.

To increase throughput, increase the configuration of CIS bulk insert array size to a larger value.

However, increasing CIS bulk insert array size causes the source Adaptive Server to use memory from the operating system for local buffers. This can lead to excessive consumption of operating system memory.

Sybase recommends that if you do choose to increase the CIS bulk insert array size default value, you do so modestly. See the CIS documentation for more information.

- CIS bulk insert array size has no effect on data throughput if the table being transferred has a text, image, or Java ADT column. When a table has a text, image, or Java ADT column in it, all data is migrated one row at a time, for the duration of the migration of that particular table. Also, no array buffering takes place.
- As the data migration is being done using CIS bulk transfer, the value for the configuration parameter CIS packet size on the source Adaptive Server can affect the speed of the data transfer. The recommended value for CIS packet size on the source Adaptive Server is the logical page size (2K, 4K, 8K, or 16K) of the target Adaptive Server.
- max packet size allowed on the target Adaptive Server should match the value of CIS packet size on the source Adaptive Server.

For more information on max packet size allowed, see the *System Administration Guide*.

- To maximize the performance of sybmigrate, increase the additional network memory configuration parameter on the target Adaptive Server to a value larger than the default.

For more information on additional network memory, see the *System Administration Guide*.

- All the above considerations affect the max memory configuration parameter. Before migrating your data, make sure that max memory is set to a sufficiently large value.
- There are three types of data that are migrated: server data, database data, and user objects. To migrate metadata (the server and database data), the target Adaptive Server must be newly installed so that the migrated metadata does not conflict with any residual data from previous usage.

If you are migrating only user objects, you can use a previously used Adaptive Server. For user data however, the target tables must be empty.

- Before migrating data, create the databases into which you want to migrate data on the target Adaptive Server. The databases should have the same name that they have on the source Adaptive Server.
- To enable conversion of character sets that do not have an internal Adaptive Server conversion routine, configure the target Adaptive Server with enable unicode conversions set to 1.

- Determine the size of the named caches and buffer pools on the target Adaptive Server. `sybmigrate` does not migrate cache configurations. You can use the information that is generated by `ddlgen` and apply it to the target Adaptive Server, or you can choose to configure larger amounts of memory, in light of the larger page size being used.

However, `sybmigrate` migrates cache bindings, therefore if the required cache is not in the target Adaptive Server, warnings are generated in the migration log.

- Before running `sybmigrate`, you must install the desired languages on the target Adaptive Server. The default language should be the same on the source and the target Adaptive Server.

If there are user messages on the source Adaptive Server that are not installed on the target Adaptive Server, `sybmigrate` aborts user message migration and reports an error.

- If you are migrating Java columns, you must enable Java on the source and target Adaptive Server prior to migration. Enter:

```
sp_configure 'enable java', 1
```

- To complete the migration, the source and target Adaptive Servers must have different local server names. Set the local server name, and then restart the servers for the change to take effect.
- If multibyte character sets are configured on the target Adaptive Server after initiating the migration process, you must manually run `dbcc fix_text` on the `sysattributes` and `sysxtypes` system catalogs to make the text columns in these catalogs consistent with the multibyte character sets.

Sybase recommends that you configure the target server character set first, and then initiate the migration process.

## Configuration and tuning for higher performance

Depending upon your server resources, you can configure `sybmigrate` and Adaptive Server for optimal performance.

## Configuration considerations for *sybmigrate*

Copy threads and create index threads are used to migrate tables and re-create indexes. When you are configuring *sybmigrate* during setup mode, the values of `COPY_THREADS` and `INDEX_THREADS` can increase the speed at which *sybmigrate* copies and migrates data.

The number of copy threads controls the number of tables for which data migration is done simultaneously. One copy thread is assigned to each table. When the thread has successfully completed one task, it moves on to another. Depending upon the size of your database and the resources for your Adaptive Server, you can increase the number of copy threads used during the migration process to improve performance.

---

**Note** When you are migrating a large number of objects in parallel, check the value of `SYBMIGRATE_MEMORY` to verify that there is sufficient memory allocated to *sybmigrate*.

---

Index threads control the number of threads used to re-create indexes on the target Adaptive Server tables. One thread per table is used to re-create the indexes. Once the indexes have been re-created on a table, the thread proceeds to the next successfully migrated table. Any threads without a task exits. The number of create index threads is expected to be substantially smaller than the number of copy threads.

If you configure `INDEX_THREADS` to a large number, be sure that the target Adaptive Server is also configured with a large number of sort buffers. The use of index threads takes up space in the target database, so make sure that the target database is configured with adequate space for the designated number of index threads. Also, you must configure the target database with extra space if you are going to be re-creating clustered indexes.

## Configuration considerations for Adaptive Server

There are several configuration parameters on both the source and target Adaptive Server that affect the performance of the migration process.

On the source Adaptive Server:

- `cis packet size` – should be equal to max page size of the target Adaptive Server.
- `number of user connections` – should be high enough to accommodate the migration of multiple tables simultaneously according to the value of `COPY_THREADS` and `INDEX_THREADS`.



- max parallel degree – should be set to a value that is larger than the largest number of partitions in a single table. Data migration is done in parallel, and if max parallel degree is not set to a value large enough to accommodate the partitioned tables, the tables do not migrate.
- number of worker processes – data migration for partitioned tables requires one worker thread per partition. Therefore, if  $t$  partitioned tables with  $p$  partitions each are migrating simultaneously, configure a total of  $t$  multiplied by  $p$  worker threads on the source Adaptive Server.
- cis bulk insert batch size – controls the number of rows after which the data transfer transaction is committed. The default value is 0. Using the default value is the safest way to ensure data integrity while migrating data, but it can result in a large number of page and row locks on the source Adaptive Server. To reduce the number of locks, increase this value.

If you increase the value of cis bulk insert batch size, only a partial data migration completes if an error occurs during the process. In this situation, manually truncate the target table and restart sybmigrate.

- cis bulk insert array size – controls the number of rows that are copied in bulk at one time. The default is 50 rows per batch. For faster data migration, increase this value.

If the table contains text or image columns, the data is transferred one row at a time, regardless of the value for cis bulk insert array size.

The following configuration parameters on the target Adaptive Server affect the performance of sybmigrate:

- max network packet size – should be set to a value that is at least equal to max page size.
- number of user connections – should be set to accommodate the migration of multiple tables in parallel and partitioned tables.

For parallel data transfer for partitioned tables, worker processes are required on the source Adaptive Server, but user connections are required on the target Adaptive Server. If you are migrating partitioned tables, set the number of user connections on the target Adaptive Server to the same value as number of worker processes on the source Adaptive Server.

- number of sort buffers – the default value of 500 is sufficient during the migration process. You can increase this value when sybmigrate rebuilds the indexes, especially if you are migrating indexes on partitioned tables.

## Possible errors to avoid

Before beginning the data migration process, sybmigrate checks for the following error conditions. If any of these conditions are detected, the migration procedure is aborted.

- A target table with existing data – any attempt to migrate data to a table that already contains data results in the failure of sybmigrate.
- A target table with existing indexes – the presence of indexes on a target table causes sybmigrate to operate in slow bcp. Manually drop all indexes before you begin the data migration.
- Unmatching numbers of partitions on the source and target tables – if the number of partitions on the source and target table do not match, the attempt to migrate data fails. sybmigrate only migrates data; it does not redistribute it across partitions.

## Auto-select dependent objects for migration

sybmigrate selects dependent objects for migration when you use the auto-select feature. The auto-select feature checks for the existence of dependent objects, and automatically migrates them to the target Adaptive Server. For a successful migration, Sybase recommends that you use this feature.

## Migrating an archive database

sybmigrate does not migrate an archive database if an entire installation is being migrated.

sybmigrate migrates an archive database only if the archive database is specifically selected for migration. When you migrate an archive database to a target server, sybmigrate automatically creates a traditional database—rather than an archive database—on the target server.

## Upgrading an Adaptive Server with an archive database

You cannot upgrade an archive database. If you load a database dump from an older version of Adaptive Server onto an archive database hosted on a newer version of Adaptive Server, the database is not internally upgraded when you execute online database.

If you upgrade an Adaptive Server containing an archive database, all the databases except the archive databases are upgraded. The archive database remains on the older version of Adaptive Server.

Sybase recommends you reload the archive database with a dump generated from an already upgraded database.

For more information about upgrading Adaptive Server, see the installation guide for your platform.

## Downgrading an Adaptive Server with an archive database

When you are downgrading to a version of Adaptive Server that does not support archive databases, be aware of the following:

- If you must downgrade an Adaptive Server containing an archive database to a version of Adaptive Server that does not support archive databases, Sybase recommends you drop the archive database before you downgrade.

To eliminate the new `sysaltusages` table, drop the scratch database before you perform the downgrade procedure. `sysaltuages` does not cause any problems if the scratch database is not dropped.

- Backup Server versions 15.0 ESD #2 and later writes a new format for compression (with `compression = compression_level`) so that the dump can be loaded into an archive database. Therefore, if you must load a compressed dump onto a version of Adaptive Server that does not support archive databases access, use the same version of Backup Server that created the compressed dump to load the compressed database dump. An earlier version of Backup Server does not support the new format of the compressed database dump.

When you are downgrading without compression, you need not worry about Backup Server at all.

## Starting `sybmigrate`

---

**Warning!** `sybmigrate` assumes that the source and target Adaptive Servers will not have any activity during the migration. If objects are created, modified, or deleted during the migration process (setup, migrate, and validate), Sybase cannot guarantee migration integrity.

---

Whether you are running the GUI or the resource file version of sybmigrate, start it with the following relevant command line arguments:

```
sybmigrate [-v ] [-h ] [-f ]
[-D 1 | 2 | 3 | 4 ]
[-l interfaces file ]
[-r input resource file ]
[-m setup | migrate | validate | report ]
[-rn status | space_est | repl | diff | password ]
[-l log file ]
[-t output template resource file ]
[-J client_charset ]
[-z language ]
[-T trace_flags ]
[-Tase trace flags ]
[-f ]
```

Where:

- -v – prints the version string and exits.
- -h – prints the help information and syntax usage and exits.
- -f overrides the locking session.

If sybmigrate exited a session inappropriately, use -f to override the source and target database binding that is created so that only one session of sybmigrate can run on a source and target database path.

- -D sets the debug level for sybmigrate. The default debug level is 2.
- -l identifies a specific *interfaces* file to find server names. If no *interfaces* file location is designated, for UNIX *SYBASE/interfaces* or for Windows *%SYBASE%\ini\sql.ini* is used.

---

**Note** You can override sybmigrate, and use the *interfaces* file by providing the -l argument if the LDAP entry is defined in *SYBASE/\$SYBASE\_OCS/config/libtcl.cfg* on Unix or in *%SYBASE%\%SYBASE\_OCS%\ini\libtcl.cfg* on Windows.

---

- -r specifies that the resource file mode is to be used in the migration process. If the input resource file is not specified by using the -r parameter, sybmigrate operates in GUI mode.

If you use the -r parameter, then you also need to use the -m argument to specify the type of operation to perform: setup, migrate, validate, or report. You can run the entire migration process in the resource file mode, or you can choose to run only parts of in this fashion.

- -m designates the types of operations that are performed:
  - setup – to set up the repository and migration working database, and to migrate the server-wide data.
  - migrate – to perform data and object migration.
  - validate – to validate the migrated objects.
  - report – to run any of the five reports. The reports can be run in the GUI and resource file mode. The available reports are:
    - status – the migrate object status report gives information about objects that have been migrated. To run this report, issue:
 

```
sybmigrate -r resource file -m report -rn status
```
    - space\_est – use the target database space estimation report to verify that you have sufficient resources allocated to your target database. In the resource file mode, issue the following command to run the space\_est report:
 

```
sybmigrate -r resource file -m report -rn space_est
```
    - repl
      - use the replication report to check any explicitly replicated objects that have been migrated, determine the type of replication system, and to produce SQL commands for users to execute on the target Adaptive Server and the Replication Server. To run the repl report, issue:
 

```
sybmigrate -r resource file -m report -rn repl
```
    - diff – checks the objects between the source and target databases. Users can run the report on individual objects, or the entire database, except for server and database information or metadata. You can run the diff report at any time. You do not need to run a setup session to run the diff report. The source and target database name do not need to be the same when running the diff report.
 

The diff report provides the following information for the following object types:

      - Server information – compares the master database system catalogs row count between the source and target Adaptive Server. This task is similar to the validation session.

- Database information – compares the user database system catalogs row count between the source and target Adaptive Server. This task is similar to the validation session.
- DDL objects – the report displays whether the objects exist on the source or the target Adaptive Servers. If the objects exists in both databases, that object is not displayed in the report.
- User table data – compares the row count of the user tables in the source and target Adaptive Server. If the table only exists in the source or target databases, the table is not displayed in the report.
- password – creates a file for the changed passwords. This report can only be run by a System Administrator.
- -m indicates what type of report to generate. If -m is not specified, all five reports are run.
- -l indicates a user-defined log file where the output of the migration process is stored. If -l is not used, the logs are stored in `$$SYBASE/$SYBASE_ASE/init/logs` or the working directory.
- -t directs sybmigrate to generate an output template resource file, to be used for subsequent migrations in the resource file mode.  
  
-t requires that you start sybmigrate using the -r argument specifying the login information. This argument also requires -m to specify what type of resource file is to be generated.

---

**Note** You can use -t only in the resource file mode.

---

- -J specifies the character set to be used for the Adaptive Server connection.
- -z specifies the language to be used for the Adaptive Server connection.
- -T sets command line trace flags. They are:
  - DUMP\_SQL – specifies that every query issued by sybmigrate is output to the log file.
  - NO\_SORTED\_DATA – overrides the default, which specifies that tables with clustered indexes are copied to the target server in order, and the clustered index is recreated using the `with_sorted_data` option.
  - LEAVE\_PTBL\_ON\_ERROR – specifies that proxy tables are not deleted on failure.

- SKIP\_CONFIG\_CHECK – specifies that configuration compatibility checks are not to be performed.
- SKIP\_PARTITION\_CHECK – specifies that partition compatibility checks are not to be performed.
- DUMP\_DDL – specifies that DDL commands are to be output to the log file.
- DUMP\_DEPEND\_OBJECT – specifies that when the auto\_select\_dependent\_objects option is used, sybmigrate outputs a list of objects added as dependents.
- ONE\_WORK\_THREAD – specifies that one work thread is to be used, overriding the current setting for schema creation threads.
- ALLOW\_DATA\_AND\_INDEX – overrides default behavior, in which indexes are created after all tables are created. Indexes are created as resources become available.
- -Tase is used to run Adaptive Server trace flags (turned on using dbcc traceon) for all Adaptive Server connections opened by sybmigrate. The trace flags should be specified in a comma-separated list.

When you run sybmigrate, there are three phases of the migration process that you must follow: setup, migrate, and validate.

## GUI mode

You can use either the GUI or the resource file mode for the migration process. You can also elect to run parts of the migration process in GUI mode, and parts of it in resource file mode.

## Setup

Before migrating data, indicate your source and target Adaptive Servers and register the paths between the source and target databases they contain. To do this, start sybmigrate with the -m setup command line option, or by selecting “Setup source databases for migration” when you are prompted in the Session Type window.

- v **Indicating your source and target Adaptive Servers and registering the paths between the source and target databases**
  - 1 The Connect to ASE window allows you to designate the source and the target Adaptive Servers for your migration process.

- You can choose from the drop-down menu in the Server fields. The menus provide a list of Adaptive Servers that are located in the default *interfaces* file (*\$SYBASE/interfaces* on UNIX or *%SYBASE%\ini\sql.ini* on Windows) or in the *interfaces* file that you specify with the *-l* command line argument.

If you are not using the *interfaces* file, you cannot use the *-l* command line argument; you must specify the source and the target Adaptive Servers in the *host:port* format.

- During the setup phase, you must be logged in to the servers as a System Administrator. Enter “sa” into the Login field, enter your password, and select Connect.

---

**Note** You can run only one session of *sybmigrate* at a time. Therefore, if there is another user running *sybmigrate* on the same source and target Adaptive Servers, you see the error message “Setup session lock: Either previous setup exit abnormal or there is another setup session running. Do you want to override?” You can override the session lock because it is possible that the previous session may have crashed or quit prematurely.

Before proceeding with the setup and migration process, verify that there are no other users running *sybmigrate*. If there is more than one user running *sybmigrate* simultaneously, Sybase cannot guarantee data integrity.

---

2 The Session Type window prompts you to select the type of operation you want to perform. Choose from:

- Setup source databases for migration
- Migrate database objects and data
- Validate the migrated objects and data
- Reports – when you select Reports, a Reports type window displays. You can choose from status, space\_est, repl, diff, or password. When you select either the space estimation or the replication report, a Report Paths Window prompts you to select the database paths on which to run the reports.

The Password, Status, and Replications reports are disabled if the setup session has not been completed between the source and target Adaptive Servers.

If you started *sybmigrate* with the *-m* option specifying setup, migrate, validate, or reports you do not see this window.



- 3 Use the Setup wizard to prepare databases for migration. The Setup wizard displays several windows:

- Choose Database window

This window prompts you to select the source and target databases located within your source and target Adaptive Servers, so that `sybmigrate` knows where to put the data from the source Adaptive Server in the target Adaptive Server.

---

**Note** The source and target databases must have identical names.

---

The Source Database drop-down list has a list of the databases in your source Adaptive Server.

The Target Database drop-down list has a list of the databases available in the target Adaptive Server. `sybmigrate` requires that you create the databases in the target Adaptive Server before beginning the migration process.

The **migration path** is a selected source and target database pair.

- Configure DDL threads

Choose the number of threads to be used to create database objects on the target server for the specified migration path.

- Configure copy threads

Choose the number of threads to be used to copy data from the source to the target for the migration path. Make sure you use sufficient numbers of threads for systems with multiple engines.

- Configure index threads

Choose the number of threads to be used to create indexes on the target server for the specified migration path. Make sure you use sufficient numbers of threads for systems with multiple engines.

You control the number of threads used for parallel table transfer. When several tables are transferred concurrently, each table requires a one-server-to-one-server CIS connection.

Suppose the data migration is performed on unpartitioned tables (each table contains a single partition). When you migrate such tables, a single server-to-server connection is established, which uses a single user connection on the source Adaptive Server and a single user connection on the target Adaptive Server.

If the data migration is performed on  $n$ -way partitioned tables, the data transfer is performed in parallel with an  $n$ -way degree of parallelism. This requires  $n$  worker processes on the source server and  $2n$  user connections on the target server.

For example, suppose you have 10  $n$ -way partitioned tables to migrate. You use four threads in `sybmigrate`, and configure the source Adaptive Server to have at least four worker processes and eight user connections. You must configure the target Adaptive Server to have at least eight user connections.

---

**Note** The value you assign to each property in the setup session becomes the default value. You can temporarily override default values in the migrate or validate session. Limit these values to the resources available to Adaptive Server.

---

- Configuring the work database

`sybmigrate` requires at least one work database during the migration process. The Database size field provides a default value in megabytes. The default value is based on the number of copy and create index threads specified in a previous window. The default is the minimum value; you can increase but not decrease it.

The Device field lets you indicate the device on which to create the work database.

---

**Note** When migrating from a source Adaptive Server version 12.0 through 12.5.0, `sybmigrate` also requires a work database on the target server. The wizard prompts for the same information for the target work database.

---

- Current paths

Review the migration paths you have selected. Right-click a migration path to display edit and delete options.

You can add paths by selecting Add Migration Path. To add paths later on, rerun `sybmigrate` in Setup mode.

- Configure repository

`sybmigrate` creates a repository database on the source server to track the migration of all migration paths. The default database size is a minimum; you can increase but not decrease it.

- Migration of server-wide data

You can choose whether or not to migrate information in system catalogs, such as login information. The options are:

- Yes – server-wide data is migrated at the end of the Setup phase.
- No – server-wide data is not migrated. You can return to this window and choose to migrate data at any time—as long as database migration has not yet begun.
- Undecided – allows you to return and choose another migration option later on. However, you cannot begin the Migration phase until you have chosen Yes or No. Undecided is useful when you want to set up the migration process, but plan to migrate data at a later date.

If the target server already has been configured for logins or other server-wide information, `sybmigrate` defaults this option to No.

The Options button provides advanced options for handling login accounts. The options let you specify:

- Whether or not to lock login accounts after migration
- How to handle login passwords when migrating across platforms
  - No change – use when migrating to the same platform (default)
  - Generate random passwords
  - Assign passwords from a list in a file
- Summary

Displays a summary of options chosen. Click Finish to perform the chosen setup tasks.

- 4 The Setup Progress window displays the progress of the setup phase.

During this time, `sybmigrate` is creating the repository database, installing the database schema, creating a working database for each selected path, and migrating the server data based on your selection, in that order. If you are running `sybmigrate` in setup mode a subsequent time, it is creating new paths for data migration. If you do not want to create new paths, there is no reason to run `sybmigrate` through the setup mode more than once.

You can to view the progress in the log by clicking Show Log. The completion of the setup process is indicated when the Current Task window displays DONE, and when the log shows SETUP\_COMPLETE. Click Close to exit the log and the Setup Progress window.

- 5 You return to the Connect to ASE window. Select Quit to exit sybmigrate. To begin the migration phase of the data migration process, exit sybmigrate and restart it in the migrate mode.

## Migrate

After you have completed setup, you are ready to begin migrating. Restart sybmigrate with the -m migrate command line option, or choose the migrate database objects and data option from the GUI window.

- 1 In the Connect to ASE window, select the source and target Adaptive Servers to which you want to connect.
- 2 If you have not started sybmigrate with the -m migrate command line argument, select the session type in the Session Type window.
- 3 The Object Selection window allows you to choose what types of database data you want to migrate.

In the Object Selection window, you can set the Copy thread, create index thread, and work thread parameters from the Setting menu bar.

In the Object Selection window, you can also request that sybmigrate Auto-Select Dependent Objects on your selected objects by right clicking the object tree node.

When you expand the database data folder, there is a file for each path that you created during setup. Each file allows you to select the data you want to migrate for that particular database. You can choose from the following:

- Database Data

---

**Note** If you choose to migrate database data, you must migrate all of it. If you deselect parts of the database data, you see an error message asking you whether or not you want to migrate database data.

If you do not migrate the server data during setup, the Database Data selection is disabled.

---

- Defaults
- Rules

- User-defined Datatypes
- Tables
- Indexes
- Referential Constraints
- Views
- Stored Procedures
- Triggers

The Status field for these objects indicates whether or not the data has successfully migrated. “Success” indicates that the data has already migrated. “Initial” means that the migration has not yet begun. If you find an error in the data that has been migrated, you can reset the Status field to Initial so that the data migrates again. The validation process acts only on those objects that have been migrated successfully, so to begin the validation process without all of the data having successfully migrated, reset the Status field to Success. “Work in Progress” means that the object was selected for migration, but that the migration was not attempted because there was some error causing sybmigrate to exit abnormally.

You can see whether or not the server data has been selected to be migrated, but this is for informational purposes only since the server data has already been migrated at this point in the migration process.

When you have selected the data that you want to migrate, click Migrate.

## Validate

The validation phase is the same as the migrate phase. The windows ask you to indicate the same information, but rather than selecting data for migration, you are selecting data for validation.

You can validate only those objects that have successfully been migrated.

## Migration and validation progress

sybmigrate keeps you informed of the migration and validation progress on the Migration/validation screen. It shows migration progress, messages reported, and a summary count of objects pending, failed, and succeeded for each task type.

You can select the Cancel button at any time, which starts a graceful shutdown of the execution progress.

## Resource file mode

You must make the following changes to the resource file mode:

- `data_copy_thread`, `create_index_thread`, and `work_thread` attributes are recognized in the setup, migration, and validate sessions of `sybmigrate`. In the setup session, these values are recorded in the repository database, and used as default values during the migrate and validate sessions. During the migrate and validate sessions, you can override the default values by specifying a new value.
- `lock_account` is a new login account management feature. `lock_account` tells `sybmigrate` to lock or unlock all accounts on the target Adaptive Server after copying the login information. Valid values are “Yes” and “No”, with “Yes” instructing `sybmigrate` to lock the target Adaptive Server accounts. To activate `lock_account`, you must set `migrate_server_data` to “Yes” in the setup session.

If the `lock_account` attribute is not set, nothing is done to target login accounts.

- `login_password_file` has been added to support changing the passwords on the target Adaptive Server. In the setup session, `login_password_file` takes the input password file or the value “<generate>”. “<generate>” is a special key used to tell `sybmigrate` to generate the passwords instead of reading them from the password file. If this attribute is not set in the resource file during the setup session, there is no change to the target Adaptive Server login passwords. To activate `login_password_file`, you must set `migrate_server_data` to “Yes” in the setup session.
- The password file must be in plain text. The content of this file consists of two columns: the login name column and the password string column. The separator between the columns are tabs and or spaces. Any lines beginning with “#” are comments.
- `auto_select_dependent_objects` is a new value that is available during the migrate and validate sessions. This attribute tells `sybmigrate` to automatically select the dependent objects for migration and validation. The valid values for this attribute are either “Yes” or “No”; “No” is the default.

- If `source_ase`, `source_ase_login`, `source_ase_password`, `target_ase`, `target_ase_login`, and `target_ase_password` attributes are not in the resource file, `sybmigrate` prompts the user for these attributes.
- In the database section of the resource file, if you do not specify any objects or SQL, all objects and types are selected.

For example, in the following resource file all object types (default, rule, table, and so on) are migrated from `pubs2` and `pubs3` databases:

```
[server]
source_ase=tho:5002
source_ase_login=sa
source_ase_password=

target_ase=tho:6002
target_ase_login=sa
target_ase_password=

[database]
source_database_name=pubs2
target_database_name=pubs2

[database]
source_database_name=pubs3
target_database_name=pubs3
```

Resource file mode is a non-interactive mode. The resource file contains all the information required for migration. You can use the resource file mode if you do not have GUI support or if you need to run batch files.

If you do not specify any object type attributes to migrate in the resource file, `sybmigrate` migrates the entire database.

If you do not specify the source or target Adaptive Server login or password in the resource file, `sybmigrate` prompts the user for this information.

Following is the format for the resource file to run `sybmigrate` in noninteractive mode. To create a resource file, type all the values into a file:

```
#
# This is a sample Migration Tool resource file.
# This resource file will migrate objects in pubs2,
# pubs3, and foo databases.
#
#####
# Server wide information
#####
```

```
[server]
# "<host name>:<port number>" or just server name.
source_ase=tho:5002
source_ase_login=sa
source_ase_password=

# "<host name>:<port number>" or just server name.
target_ase=tho:6002
target_ase_login=sa
target_ase_password=

# Repository database setup attributes. This is
required with "setup" mode.
# Repository database size in MB.
repository_database_size=7
# Device used to create the "sybmigrate" database.
repository_device=master

# Migrate server wide data - logins, roles, remote servers, etc...
# valid only with "setup" mode, default is yes
migrate_server_data=yes

# Tell sybmigrate to lock or unlock all login accounts on the
# target Adaptive Server. Valid values are "yes" and "no":
# "yes" to lock and "no" to unlock. This is only valid if
# "migrate_server_data" is set to "yes" and run in "setup" mode.
# If this attribute is not specified, target Adaptive Server login
# accounts are not change.
lock_account=no

# Change target Adaptive Server login passwords. This is only valid
# if "migrate_server_data" is set to "yes" and run in "setup" mode.
# If this attribute is not specified, target Adaptive Server login
# accounts are not change.
# The valid values are "<generate>" and password file.
# "<generate>" instructs sybmigrate to use random passwords.
# Password file instructs sybmigrate to use the passwords from
# this file.
# The content of the password file consists of two columns:
# the login name column and the password string column.
# The separator between the columns are tabs and or spaces.
login_password_file=<generate>

#####
# Database information
#####
```



```
#
# Migrate the "pubs2" database objects
#
[database]
# Specify the source target database to migrate.
source_database_name=pubs2
target_database_name=pubs2

# Migrate database data, valid only if "migrate_server_data"
# was set to "yes" in "setup" mode. This is default to yes.
migrate_database_data=yes

# Work database setup attributes. This is required with "setup" mode.
# Work database size in MB.
work_database_size=5
# Device used to create the work database.
work_database_device=master

# Number of threads use to do user table data copy
data_copy_thread=5

# Number of thread use to create indexes.
create_index_thread=1

# Number of thread use to do ddl migration/validation
work_thread=10

# Automatically select the dependent objects for migration and
# validation. Valid values are "yes" or "no".
auto_select_dependent_objects=yes

#
## Migrate objects
#
# These attributes specify the list of DDL object to
# migrate or validate. User can directly specify the
# list of DDL object or ask Migration tool to query the
# list. Directly specifying the list has the higher
# precedence. The SQL command will ignore if the list
# is given.
#
# Note:
# * The SQL command for the "*_list_from_sql" attributes
# must return column <object name> or columns <user
# name> and <object name>
# * Index type must also specify the table name. For
```

```
# example, "<table>.<index name>" for
# "index_create_list" attribute or columns <table>,
# <index name> for "index_create_list_from_sql"
# attribute.
# * Value "<ALL_OBJECTS>" can be used on any of the
# attributes to specify all objects for the type.
# * If none of these attributes are given, all objects
# and data are migrated.
#
user_defined_type_create_list=
id
dbo.tid

default_create_list_from_sql=
select user_name(uid), name from sysobjects
where type = 'D'

rule_create_list=
pub_idrule, title_idrule

table_create_list=
publishers
titles
dbo.authors
dbo.titleauthor
dbo.roysched
stores
dbo.sales
dbo.salesdetail
dbo.discounts
dbo.au_pix
blurbs

table_migrate_list=
dbo.publishers titles dbo.authors dbo.titleauthor
dbo.roysched
stores dbo.sales dbo.salesdetail dbo.discounts au_pix
dbo.blurbs

index_create_list=
dbo.authors.auidind
dbo.authors.aumind
publishers.pubind
roysched.titleidind
sales.salesind
salesdetail.titleidind
```

```
salesdetail.salesdetailind
titleauthor.taaind
titleauthor.aidind
titleauthor.titleidind
titles.titleidind
titles.titleind

trigger_create_list=
delttitle
totalsales_trig

store_procedure_create_list_from_sql=
select name from sysobjects where type = 'P'

view_create_list_from_sql=<ALL_OBJECTS>

referential_constraint_create_list_from_sql=<ALL_OBJECTS>

logical_key_create_list_from_sql=<ALL_OBJECTS>

#####

#
# Migrate the "pubs3" database objects
#
[database]
source_database_name=pubs3
target_database_name=pubs3

# Migrate database data - user, etc.
migrate_database_data=yes

# These two attributes valid only with "setup" mode
work_database_size=5
work_database_device=master

# Number of threads use to do user table data copy
data_copy_thread=5

# Number of thread use to create indexes.
create_index_thread=1

# Number of thread use to do ddl migration/validation
work_thread=10

# Migrate objects
```

```
user_defined_type_create_list=<ALL_OBJECTS>

default_create_list=<ALL_OBJECTS>

rule_create_list=<ALL_OBJECTS>

table_create_list=
dbo.authors
publishers
dbo.titles
dbo.roysched
stores
dbo.sales
dbo.store_employees
salesdetail
dbo.titleauthor
dbo.discounts
blurbs

table_migrate_list_from_sql=<ALL_OBJECTS>

index_create_list=<ALL_OBJECTS>

trigger_create_list=<ALL_OBJECTS>

store_procedure_create_list=<ALL_OBJECTS>

view_create_list=<ALL_OBJECTS>

referential_constraint_create_list_from_sql=<ALL_OBJECTS>

logical_key_create_list_from_sql=<ALL_OBJECTS>

#####

#
# Migrate all the "foo" database objects with default settings.
#
[database]
source_database_name=foo
target_database_name=foo

# Migrate database data - user, etc.
migrate_database_data=yes

# These two attributes valid only with "setup" mode
```

```
work_database_size=5
work_database_device=master

# Number of threads use to do user table data copy
data_copy_thread=5

# Number of thread use to create indexes.
create_index_thread=1

# Number of thread use to do ddl migration/validation
work_thread=10
```

## Sybmigrate with Encrypted Columns

For databases with encrypted columns, sybmigrate:

- 1 Migrates the system encryption password. If you specify not to migrate the system encryption password, sybmigrate migrates the encrypted columns using the reencrypt method instead of migrating the ciphertext directly.
- 2 Migrates the encryption keys. You may select the keys to migrate. sybmigrate automatically selects keys in the current database used to encrypt columns in the same database. If you have selected migration of the system encryption password, sybmigrate migrates the encryption keys using their actual values. The key values from the *sysencryptkeys* system table have been encrypted using the system encryption password and these are the values that are migrated. If you have not migrated the system encryption password, sybmigrate migrates the keys by name, to avoid migrating keys that will not decrypt correctly at the target. Migrating the key by name causes the key at the target to be created with a different key value from the key at the source.
- 3 Migrates the data. By default, the data is transferred in its ciphertext form. Ciphertext data can be migrated to a different operating system. Character data requires that the target server uses the same character set as the source.

sybmigrate works on a database as a unit of work. If your database on the source server has data encrypted by a key in another database, migrate the key's database first.

sybmigrate chooses to reencrypt migrated data when:

- Any keys in the current database are specifically not selected for migration, or already exist in the target server. There is no guarantee that the keys at the target are identical to the keys at the source, so the migrating data must be reencrypted.
- The system password was not selected for migration. When the system password at the target differs from that at the source, the keys cannot be migrated by value. In turn, the data cannot be migrated as ciphertext.
- The user uses the following flag:

```
sybmigrate -T 'ALWAYS_REENCRYPT'
```

Reencrypting data can slow performance. A message to this effect is written to the migration log file when you perform migration with reencryption mode.

To migrate encrypted columns, you must have both `sa_role` and `sso_role` enabled.

## Post-migration activities

- `sybmigrate` supports the migration of only the objects listed elsewhere in this document. Manually migrate other schema objects and configuration information to ensure the target Adaptive Server is fully functional.
- Statistics for indexes are automatically re-created when you rebuild the indexes. However, `sybmigrate` does not re-create statistics from non-index columns. Any user-defined step values for index statistics are not retained during migration. To obtain target-server-side statistics similar to the source-server-side statistics, use `optdiag` to identify the tables with non-index columns that include statistics. Once you have determined which non-index columns include statistics, update the statistics manually.
- Any message requiring user attention preceded by the word “attention” and logged in the migration log.
- Run the object migrations status report to verify that all objects have been migrated.

## Migrating databases in the Replication Server domain

The Replication Server domain includes one or more:

- Primary databases
- Replicate databases
- Replication Server System Databases (RSSDs)

---

**Note** The RSSD stores Replication Server system tables; in addition, it can also be a primary or a replicate database.

---

You can migrate any of these databases, but the process requires additional steps to ensure success.

### Premigration procedures

Make sure that replication from or into each database is complete before initiating migration. This means that:

- For a primary database – all changes have been applied to all subscribing databases
- For a replicate database – all changes to which the database subscribes have been applied

---

**Note** All transactions in the Replication Server inbound and outbound queues must be applied. After migration, there is no way to restore data left in the Adaptive Server transaction log.

---

#### v Before migrating data

1 Log in to the Replication Server and suspend log transfer. Enter:

```
suspend log transfer from server.database
```

2 Log in to the Adaptive Server, and shut down the RepAgent. Enter:

```
use database
sp_stop_rep_agent database
```

3 Suspend all DSI connections to the replicate database. Log in to the Replication Server and enter:

```
suspend connection to server.database
```

- 4 Put the Replication Server in hibernation mode. Enter:

```
sysadmin hibernate_on, replication_server
```

Before starting the migration process, sybmigrate records replication information in its log. The information needed to restore the replication information during the postmigration steps can be retrieved from this log. See “Postmigration procedures” on page 294 for more information.

## Postmigration procedures

After migration, restore the replication information in the database. These steps can be generated by the repl report.

If the page size changes between the source and target, you must also follow directions in “Amending system tables when the logical page size changes” on page 296.

Replication Server identifies all connections by *server\_name.database\_name*. After migration, you must change the name of the target server (the server you are migrating to) to that of the source server (the server you are migrating from).

## Restoring primary databases

Follow this procedure for all primary databases, including the RSSD, if it is a primary database.

If the page size changes during the migration, you must also alter the *rs\_lastcommit* and *rs\_threads* system tables as described in “Amending system tables when the logical page size changes” on page 296.

### v Restoring primary databases

- 1 If the original primary database had warm standby on, restore the standby status. Enter:

```
sp_reptostandby database_name, status
```

sybmigrate saves the standby status in the migration log of the source database.

- 2 Increase the generation ID by 1. Enter:

```
dbcc settrunc ("ltm", "gen_id", gen_id)
```



You can view the current generation ID in the migration log of the source database.

- 3 Reset the secondary truncation point:

```
dbcc settrunc ("ltm", "valid")
```

- 4 Zero the Replication Server locator value for this database. Enter:

```
rs_zeroltm server, database_name
```

- 5 If this database is an active connection in a warm standby configuration, rematerialize the standby database by dumping the primary and loading the dumps into the standby. See the Replication Server documentation for instructions.

- 6 Start the RepAgent on the primary database. Enter:

```
sp_start_rep_agent database_name
```

- 7 Log in to the Replication Server and restart log transfer:

```
resume log transfer from server.database
```

## Restoring replicate databases

If the page size does not change during migration, there are no postmigration steps necessary for replicate databases.

If the page size does change, follow the procedure in “Amending system tables when the logical page size changes” on page 296.

## Restoring the RSSD

This is the procedure for restoring the RSSD.

### v Restoring the RSSD

- 1 If the RSSD is a primary database, follow the procedure in “Restoring primary databases” on page 294.

If the page size changes, make sure you alter the `rs_lastcommit` and `rs_threads` system tables as instructed.

- 2 Turn off hibernation for the Replication Server. Log in to Replication Server and enter:

```
sysadmin hiberate_off replication_server
```

## Amending system tables when the logical page size changes

If the logical page size changes during migration, you must alter the `rs_lastcommit` and `rs_threads` system tables to account for the change.

Follow this procedure for **all** databases in which the page size has changed.

### v Altering the `rs_lastcommit` and `rs_threads` system tables

1 Alter the `rs_lastcommit` table. Enter:

```
declare @pad8_size integer
declare @alter_cmd varchar(200)

select @pad8_size = (@@maxpagesize / 2)
- (select sum(A.length) from
syscolumns A, sysobjects B
where A.id = B.id
and B.name = 'rs_lastcommit')
+ (select A.length from
syscolumns A, sysobjects B
where A.id = B.id
and B.name = 'rs_lastcommit'
and A.name = 'pad8')

select @alter_cmd = "alter table rs_lastcommit "
+ "modify pad8 char("
+ convert(varchar(100), @pad8_size)
+ ")"
execute (@alter_cmd)
go
```

2 Alter the `rs_threads` table. Enter:

```
declare @pad4_size integer
declare @alter_cmd varchar(200)

select @pad4_size = (@@maxpagesize / 2)
- (select sum(A.length) from
syscolumns A, sysobjects B
where A.id = B.id
and B.name = 'rs_threads')
+ (select A.length from
syscolumns A, sysobjects B
where A.id = B.id
and B.name = 'rs_threads'
and A.name = 'pad4')

select @alter_cmd = "alter table rs_threads "
```

```

+ "modify pad4 char("
+ convert (varchar(100), @pad4_size)
+ ")"
execute (@alter_cmd)
go

```

## Logs

In the migration tool log, information about replicated objects is preceded by the following banner:

```
=== Replication Information for Database 'pdb1' ===
```

The following is a sample log file for a primary database named pdb1:

```
sp_repostandby 'pdb1' is NONE.
```

If the standby status for the database is not NONE, use the standby status as described in the post-migration steps above.

```
sp_config_rep_agent 'pdb1'
```

sp\_config\_rep\_agent requests the current RepAgent configuration. The migration tool automatically restores RepAgent configuration, and you can use this log to verify the RepAgent configuration.

Parameter name	Default	Config Value	Run value
priority	5	5	5
fade timeout	30	30	30
scan timeout	15	15	15
retry timeout	60	60	60
rs username	n/a	rs1_user	rs1_user
trace flags	0	8194	8194
batch ltl	true	true	true
rs servername	n/a	rs1	rs1
send buffer size	2k	2k	2k
trace log file	n/a	n/a	n/a
connect database	n/a	pdb1	pdb1
connect dataserver	n/a	pds1	pds1
can batch size	1000	1000	1000
security mechanism	n/a	n/a	n/a
msg integrity	false	false	false
unified login	false	false	false
kip ltl errors	false	false	false
msg origin check	false	false	false
short ltl keywords	false	false	false
msg confidentiality	false	false	false
data limits filter mode	stop	stop	stop

```

msg replay detection          false          false          false
mutual authentication         false          false          false
send structured oids         false          false          false
send warm standby xacts     false          false          false
msg out-of-sequence check   false          false          false
skip unsupported features     false          false          false
send maint xacts to replicate false          false          false
(28 rows affected)

```

This is a list of explicitly replicated tables. `symigrate` automatically restores the replication status for explicitly replicated tables, and you can use this part of the log to verify the replication status of explicitly replicated tables.

```

sp_setreptable
Name                                Repdef Mode
-----
t1                                  owner_off
t2                                  owner_on
(2 rows affected)

```

This is a list of explicitly replicated stored procedures. The migration tool automatically restores the replication status for explicitly replicated stored procedures, and you can use this part of the log to verify the replication status of explicitly replicated stored procedures.

```

sp_setrepproc
Name                                Type          Log Mode
-----
p1                                  function     log_sproc
p2                                  function     log_current
p3                                  table        log_sproc
p4                                  table        log_current
(4 rows affected)

```

This is information about the secondary truncation page. You will need the `generation_id` column during the post-migration steps.

```

dbcc gettrunc
secondary_trunc_page  secondary_trunc_state  dbrepstat  generation_id
      database_id      database_name          ltl_version
-----
621                    1                    167         0
      6                    pdb1                   400
(1 row affected)

```

This appears to be a replicated primary database.  
Make sure the post processing steps for a replicated primary database are performed. Please consult the manuals for the steps that need to be performed.

The following is an example log entry if your database is a replicate database.

This appears to be a replicate database.  
If the pagesize is greater than 2K, make sure the post processing steps for a replicate database are performed. Please consult the manuals for the steps that need to be performed.

The following is an example log entry for an RSSD database.

This appears to be a replication system database  
Make sure the post processing steps for a replication system database are performed. Please consult the manuals for the steps that need to be performed

All three logs can be present for a database, since a database can list the three categories.

## Migrating databases that support wide data

Adaptive Server version 12.5 and later can generate data wider than what Replication Server version 12.1 and earlier can handle. If RepAgent passes wide data to Replication Server 12.1 or earlier, Replication Server threads may shut down.

RepAgent communicates with Replication Server using Log Transfer Language (LTL). When the RepAgent connects to Replication Server, it returns an LTL version as shown in Table 8-1.

**Table 8-1: Replication Server and LTL versions**

Replication Server version	LTL version
12.1 and earlier	< 400
12.5 and later	>= 400

If Replication Server returns an LTL version less than 400, RepAgent uses the setting of the data limits filter mode option to determine how to treat wide data.

You can set the data limits filter mode option using `sp_config_rep_agent`. Values for data limits filter mode are:

- stop – RepAgent shuts down when it encounters data too wide for Replication Server to process (the default when the LTL version is less than 400).
- skip – RepAgent ignores data too wide for Replication Server to process, and logs an informational message.
- truncate – RepAgent truncates wide data so that Replication Server can process it. If the table or stored procedures has more than 250 columns or parameters, only the first 250 columns or parameters are sent. If the column or parameter is wider than 255 bytes, only the first 255 bytes are sent.
- off – RepAgent sends wide data to the Replication Server; Replication Server threads may shut down.

Table 8-2 shows column and width limits for Replication Server 12.1 and earlier and Replication Server 12.5 and later.

**Table 8-2: Replication Server column number and width limits**

Property	Replication Server 12.1 and earlier	Replication Server 12.5 and later
Column count	250	65535
Column width	255	65535

## Limitations

---

**Note** When migrating server data, sybmigrate requires that the target Adaptive Server catalog contain only default data. Default data on Windows machines is different from UNIX machines. This causes problems when migrating from UNIX to Windows machines. To successfully migrate from a UNIX machine to a Windows machine, delete the XP Server name and the mon\_user login on the target Windows machine.

---

### High availability

Data migration is not supported while you are in high availability. You must stop high availability before beginning database migration.

- v **Stopping high availability before beginning database migration**
  - 1 Decouple primary and secondary Adaptive Servers.

- 2 Migrate primary source Adaptive Server and secondary source Adaptive Server data to the primary target Adaptive Server and secondary target Adaptive Server separately.
- 3 Configure the target Adaptive Server for high availability.

---

**Warning!** The primary and the secondary Adaptive Servers must be configured to the same logical page size to run high availability.

---

#### Other limitations

- sybmigrate does not do any special processing for a DTM/XA environment. The status of open transactions and outstanding prepared transactions should be given consideration. If any special handling is required, you must do it manually.
- There is no reliable way for sybmigrate to determine the dependency of various objects. sybmigrate does not attempt to create an order in which objects are migrated based on their dependencies on other objects. Views can be dependent upon other views, and they will not be re-created if the view on which they are dependent has not yet been migrated. The migration of stored procedures and triggers may not be successful if the data on which they depend has not yet been migrated. Cross-database dependencies mean that you need to coordinate the migration of related objects. If dependencies are within the selected set, sybmigrate takes care of those dependencies. However, if dependencies exist outside the selected set, you may need to run sybmigrate through migration more than one time. For this reason, you may need to perform some partial retries to successfully complete the data migration.
- Adaptive Server versions 12.5.3 and later allow you to specify the size and location of a work database on your target server. When migrating a database or server from a source server with Adaptive Server Enterprise versions 12.0 and later but earlier than 12.5.0.1, you must specify the size and location of a work database on the target server.
- The name of the source and the target databases must be the same. SQL schema generated by ddlgen may have objects that must be qualified with the source Adaptive Server name.
- sybmigrate does not support any kind of auditing for migration activities.
- When renaming any of the compiled objects (procs, views, rules, defaults) the object name in syscomments is not updated.

During the migration, the DDLGen query the object from syscomments with the old name in the text. This old name in the text causes problems for sybmigrate during the DDL migration.

## Troubleshooting and error messages

This section discusses common errors and how to address them, as well as different error messages and their meaning.

### Objects fail to migrate

Objects often fail to migrate on the first attempt. `sybmigrate` automatically retries all failed migration attempts. However, if you choose to migrate an object that is dependent upon another object that is not migrated, the migration fails.

To prevent failed migration of objects, examine the dependencies of objects that you select for migration. For example, you cannot migrate a trigger if the table on which the trigger is defined is not also migrated. Similarly, views can be created on other views or tables, and if these objects are not migrated, the migration of the view fails.

### Beginning database migration

When you are in the setup phase of the migration process, you are asked to decide whether or not you want to migrate server data. You must select from yes, no, or undecided.

“Undecided” provides you with the flexibility of setting up the migration process, but being able to return to the process at a later date that is more convenient for migration. If you select Undecided, you cannot begin the database migration until you indicate whether you want to migrate server data.

If you indicate that you do not want to migrate server data during setup, you cannot migrate database data during migration. You can override this limitation in GUI mode.

### “Connection refused” and “Unable to obtain connection to the server”

There are two possible reasons why you may encounter these error messages.

- If either the source or the target Adaptive Server is not running, `sybmigrate` cannot establish a connection.



- The number of user connections configuration parameter must be configured to provide sufficient resources on both the source and target Adaptive Servers.

## Target server cannot be reached from source server

The interfaces file is used to start the source Adaptive Server. Verify that it has an entry that identifies the target Adaptive Server.

Verify that your login can access the target Adaptive Server from the source Adaptive Server.

## If *sybmigrate* hangs during migration

If *sybmigrate* hangs during the migration process, check the *sybmigrate* log in `$SYBASE/$SYBASE_ASE/init/logs` for any errors or exceptions.

Also, check your Adaptive Server logs. If the Adaptive Server logs run out of space on the database, increase the database size, and install the `sp_thresholdaction` stored procedure to do dump tran when the log is full.

## Merging two databases

To merge two databases on the source Adaptive Server into one database on the target Adaptive Server, use the following procedure.

- v **Merging two databases**
  - 1 Set up and migrate the first database.
  - 2 After migrating the first database, rename the target database so that it has the same name as the second source database.
  - 3 Set up and migrate the second database.

---

**Note** You cannot migrate the database data for the second database because the users, roles and other database data already exist on the target database. You can still migrate user data.

---

## Post-migration failure cleanup

If sybmigrate fails unexpectedly, rerun sybmigrate on the areas that failed. If it fails again with no more progress, clean up the source and target Adaptive Servers, and begin migration again. There are actions that you must perform on both the source and target Adaptive Server.

On the source Adaptive Server:

- Drop the temporary working databases mtpdb\$%.
- Drop the repository database sybmigratedb.
- Drop all remote servers mtrs\$%.

On the target Adaptive Server:

- If server data was migrated, rebuild the target Adaptive Server with srvbuild or syconfig.
- Re-create the target databases.

## Remigrating one database

To remigrate a specific database, you must:

- 1 Start sybmigrate.
- 2 In the Setup Paths window, during the setup session, right-click the migration path you want to redo.
- 3 Select Delete Migration Path on the pop-up menu.
- 4 Clean up or remove the migrated data and objects on the target database, or drop and re-create the target database.
- 5 Restart sybmigrate and run it from setup mode.

## Re-creating an individual object

To re-create an individual object:

- 1 In the target Adaptive Server, drop the object you want to re-create.
- 2 Start sybmigrate in the migration session, and go to the Migrate Object Selection window. Highlight the object you want to create and right-click.
- 3 From the pop-up menu, select Reset Object to Initial status.

- 4 Complete the migration process.

## Connection fail

If you receive a connection fail error message even though the source and target Adaptive Servers are running, you may be using the wrong character set. When you are using `sybmigrate`, you must use the default character set. Run `sybmigrate` with the `-J charset` option, to change the character set you are using.

## “Insufficient memory in JVM shared class”

If you see the following error in the server log, it indicates that you must reconfigure the size of shared class heap configuration parameter to a larger value.

```
01:00000:00036:2002/01/28 14:17:05.63 server Java VM
Host: Memory allocation request failed because of
insufficient memory in Jvm Shared Class.
```

## “There is not enough memory in the procedure cache”

If you see the error message `there is not enough memory in the procedure cache` during the migration of indexes, use `sp_configure procedure cache size` to increase the procedure cache.

## java.lang related error

If you receive  
`java.lang.NoClassDefFoundError: com/sybase/jdbcx/SybDriver`  
when you are connecting to Adaptive Server, check to make sure you have `jConnect 6.0` installed in your `$SYBASE` directory (`$SYBASE/jConnect-6_0`).



# Index

## Symbols

!! (exclamation points) operating system commands  
    prefix (isql) 210  
::= (BNF notation)  
    in SQL statements xv  
, (comma)  
    in SQL statements xv  
{ } (curly braces)  
    in SQL statements xv  
( ) (parentheses)  
    in SQL statements xv  
[ ] (square brackets)  
    in SQL statements xv  
< (redirect in) in isql 15  
> (redirect out) in isql 15  
\  
    escaping special characters 116  
\  
    escaping special characters 116  
\  
    data field terminator in interactive bcp 69  
\  
    null character terminator in interactive bcp 69  
" " (enclosing special characters) 116

## A

Adaptive Server  
    configuration of, for migration 270  
    executing (dataserver) 156  
    executing (sqlsrvr) 237  
    rolling back processes 84  
    upgrading (sqlupgrade) 227, 244  
    upgrading with resource files (sqlupgraderes)  
        245  
adding  
    network transport addresses in dscdit 99, 101  
    server entries in dscp 108  
    server entries in dscp 96, 98  
**additional network memory** parameter 268  
allow\_dup\_row option to create index, and bcp 88

application programs, copying data from 76  
archive database access  
    downgrading 273  
    migrating 266, 272  
ASCII format, bcp and 60  
ascii\_7 character set  
    defncopy and 187

## B

backing up, compared to bulk copying 90  
backslash (\)  
    escaping special characters 116  
    terminator in interactive bcp 69  
Backup Server  
    See also backupserver utility command 122  
backup Server 122  
backupserver utility command 122  
    character set, default 123  
    defined 122  
    error log file 123  
    error messages in 127  
    full path names, specifying 123  
    interfaces file 126  
    interfaces file, name and location of 122  
    language, default 123  
    LC\_ALL environment variable 123  
    permissions required for 127  
    server connections, number of 122  
    server names, specifying 122  
    starting servers and 126  
    trace flags 126  
Backus Naur Form (BNF) notation xv  
-bbatch\_size parameter 84  
bcp utility command 90, 128  
    allow\_dup\_row option to create index 88  
    ASCII format and 60  
    batch operations in 84  
    batch size settings 131

- bbatch\_size parameter 84
- binary format and 43
- character format 60
- character format files 60
- character format, default 61
- character formats accepted 43
- character set defaults 134
- configuration parameters 87
- copying data in 50, 53, 79, 83
- copying data out 76, 79, 86
- copying in batches of table rows 84
- data integrity 89
- data loss and dumping 49
- data storage size in 64
- data transfer, preparing for 44
- datatypes 69
- default values for data 89
- defaults for columns and datatypes 89
- defined 128
- described 42
- dump database and 90
- error files 87
- errors in data conversion 87
- examples for using 133
- fast version of 48
- fast version of, and data recoverability 47
- field lengths 60
- field terminators 61, 64
- file storage types in 63
- format files 72, 76
- IDENTITY columns and 58
- improving performance of operations 87
- improving recoverability when copying data in 85
- index creation 88
- indexes and triggers, dropping 50
- insert and 90
- interactive mode 62, 70
- languages, using alternate 83
- load database and 90
- load transaction and 90
- maximum speed, enabling 89
- native file format and 60
- native format option and 60
- non-character to character datatype default field lengths 67
- non-interactive 60
- non-iso\_1 data files and 83
- other Adaptive Server utilities and 89
- page allocations, increasing for 53
- partitioned tables and 50, 59, 86
- password encryption in 135
- performance issues with 46, 58, 89
- permissions required for 43, 66
- prefix length 64
- prompts and responses. *See* Interactive bcp 62
- prompts in 63
- rolling back processes 84
- row terminators 61
- rules and copying data 89
- select into/bulkcopy/plsort and 49
- slow version and deadlocks on index pages 52
- slow version of 48
- sp\_dboption 49
- space needed for data 50
- special characters, handling 132
- speed, and indexes and triggers 47
- speed, modes of 45
- storage types 69
- system data format (SDF) output in 77
- system differences, operating 61
- table defaults and copying data 89
- terminators used in 43
- transferring data between programs 42
- triggers and data copying 89
- triggers not fired on target table 47
- unique IDENTITY column values and 58
- using with row-level access rules 84
- warning about data recovery 131
- binary data
  - bcp and 43
  - default in interactive bcp 66
- BNF notation in SQL statements xv
- brackets. *See* square brackets [ ]
- buffer, query 12
- buildmaster utility command. *See* dataserver
- bulk copy process
  - See also* bcp utility command 128

## C

carriage-return data field terminator (`\r`) in interactive bcp 69

case sensitivity  
in SQL xvi

char datatype, and bcp 60

character formats  
bcp 43  
default in bcp 60, 61  
terminators for 69

character set  
specification for migration 276

character sets  
backserver, default in 123  
bcp, defaults in 134  
converting from non-character data 67  
installing and modifying (sqllocres) 236  
installing and modifying, in GUI (sqlloc) 235  
loading, with charset 153  
platform default 83

charset utility command 153  
defined 153  
permissions required for 153  
settings for 153

**CIS bulk insert array size**  
configuration for migration 267, 271

**CIS bulk insert batch size** 271

**CIS packet size**  
configuration for migration 270  
for migration 268

closing a session  
dscp 106  
dsedit 94

column precision in numeric or decimal storage formats 76

column scale in numeric or decimal storage formats 76

columns  
datatype sizes and 67  
default values and bcp 89  
fixed- and variable-length 66  
null 87  
separator character (isql) 11

comma (,)  
in SQL statements xv

comma-delimited output 77, 78

command terminator (isql) 10  
statistics option interaction 14

component directory for migration 265

conventions  
*See also* syntax  
Transact-SQL syntax xv  
used in the Reference Manual xiv

copy threads on **sybmigrate** 270

copying  
between sessions in dscp 110  
different sessions in dscp 110  
new entries with dscp 110  
server entries with dscp 109  
server entries with dsedit 98, 99, 100, 101

copying data in  
batch operations in 84  
improving recoverability after rolling back 85  
parallel bcp, requirements for 53, 58  
partitioned tables 51  
partitions, random use of 51  
steps, using fast version of bcp 50

copying data in with interactive bcp 79, 83  
compatibility of datatypes, and failure 82  
delimiters 82  
error files and 88  
field lengths 80

copying data out with interactive bcp 76, 79  
delimiters 78  
error files and 88  
fixed-length fields 77  
for other software 76  
text and image data 86

copying definitions 186

create index command, bcp and duplicate rows 88

creating  
new servers (srvbuild) 246, 247  
new servers using resource files (srvbuildres) 248

curly braces ({} ) in SQL statements xv

## D

data  
changing with Adaptive Server commands 90  
conversion errors 87, 88  
importing and exporting with bcp 42

## Index

- moving with Adaptive Server commands 89
- padding with spaces in interactive bcp 66
- parsing. See Field terminators 61
- permission required to copy into tables 43, 140
- recoverability 47
- transferring from different programs using bcp 42
- data transfer, default formats with bcp 60
- database data, migration of 261
- database management systems, other 76
- database objects
  - copying using bcp 128
- databases, copying with bcp 90
- dataserver utility command 156
  - defined 156
  - passwords, generating new 160
  - permissions required for 161, 241
- datatypes
  - bcp file storage types for 62, 65
  - bcp format files for 72
  - bcp, used in 69
  - char 60
  - copying and compatibility 82
  - default values and bcp 89
  - field lengths in interactive bcp 64, 70
  - implicit conversions of 64, 65
  - non-character to character default field lengths in bcp 67
  - storage (SYB) 74
- debug level for **sybmigrate** 274
- debugging utility 230
- default network packet size configuration parameter 87
- defaults
  - bcp data conversion 67
  - bcp prompts 62, 70
  - character sets in backupserver 123
  - character sets, in bcp 134
  - copying into tables using data 89
  - copying, with defncopy 186
  - languages in backupserver 123
  - prompts in interactive bcp 62, 70
  - select into/bulkcopy/pllsort option settings in new databases 50
- definitions
  - backupserver utility command 122
  - bcp utility command 128
  - charset utility command 153
  - copying. See defncopy utility command 186
  - dataserver utility command 156
  - defncopy utility command 186
  - dscp utility command 192
  - dsedit utility command 193
  - isql utility command 201
  - optdiag utility command 216, 223
  - showserver utility command 229
  - sqlloc utility command 235
  - sqllocres utility command 236
  - sqlsrvr utility command 237
  - sqlupgrade utility command 244
  - sqlupgraderes utility command 245
  - srvbuild utility command 246
  - srvbuildres utility command 248
  - startserver utility command 249
  - sybmigrate utility command 252
  - xpserver utility command 257
- defncopy utility command 186
  - ascii\_7 character set and 187
  - defined 186
  - encrypted text 191
  - examples 189
  - failure due to long comments 190
  - passwords and crashing 186
  - Report Workbench, incompatibility with 186
- deleting
  - server entries (dscp) 111
- delimiters
  - copying data in with 82
  - copying data out with 78
- directory services
  - entries, adding 96
  - entries, copying to 98, 100, 101
  - entries, deleting 97
  - entries, modifying 96
  - entries, renaming 97
  - opening to, with dsedit 93
- DISPLAY environment variable
  - setting, in dsedit 92
- downgrading an archive database 273
- dscp utility command 192
  - commands, table of 112
  - defined 105, 192
  - entries, creating new, by copying 110
  - examples of 192



- exit command 112
- help 105, 112
- permissions required for 192
- quit command 112
- server attributes 107
- server entries, adding 108
- server entries, copying 109
- server entries, deleting 111
- server entries, listing 110
- server entries, modifying 108
- server entry contents, viewing 111
- sessions, closing 106
- sessions, copying between 110
- sessions, copying to different 110
- sessions, listing 106
- sessions, opening 106
- starting 105
- dsedit utility command 193
  - about 91
  - command line arguments 91
  - defined 193
  - Directory Service Session screen 94
  - DISPLAY environment variable and 92
  - does not start 102
  - libtcl.cfg file 93
  - network transport addresses, adding 99, 101
  - network transport addresses, editing 99
  - opening with sql.ini 93
  - permissions required for 92, 193
  - remote machines, running from 92
  - Select a Directory Service screen 92
  - server attributes 95
  - server entries, adding 96, 98
  - server entries, cannot add, modify, or delete 102
  - server entries, copying 98, 99, 100, 101
  - server entries, deleting 97
  - server entries, modifying 97, 98
  - server entries, renaming 97
  - sessions, closing 94
  - sessions, opening 94
  - starting 92
  - starting from command prompt 91
  - starting from Windows Explorer 91
  - SYBASE environmental variable and locating
    - libtcl.cfg 93
  - troubleshooting 91, 102
  - xd2 Unable to open X displayxd3 error message 102
  - DSLISTEN environment variable
    - backupserver and 126
  - dump database command
    - bcp and 90
  - dump transaction command
    - bcp and 90
    - error message recommending use of dump database 47
- E**
- echo input (isql) 15
- editing
  - interfaces files 92, 94
  - interfaces files in GUI (dsedit) 193
  - network transport addresses in dsedit 99
- editing interfaces files in GUI
  - See also dsedit utility command 193
- enable unicode conversions** for migration 268
- encrypted hidden text 191
- environment variables
  - DSLISTEN (backupserver) 126
  - LANG (backupserver) 123
  - LC\_ALL (backupserver) 123
  - release path, for **sybmigrate** 265
  - SYBASE\_ASE 265
  - SYBASE\_JRE environment variable 265
- error files
  - bcp and 87, 88
  - copying out and 88
- error log files
  - for backupserver 123
- error messages
  - dsedit and X display 102
  - dump database use over dump transaction 47
  - in backupserver 127
  - select into/bulkcopy/pllsort in fast bcp on tables with
    - no indexes or triggers 49
- error messages for migration 302
- errors
  - in data conversions 87, 88
  - operations, saving during copy in 87
  - operations, saving during copy out 88

## Index

### examples

- bcp utility command 133
- defncopy utility command 189
- dscp utility command 192
- isql utility command 202
- showserver utility command 229
- startserver utility command 249, 250

exclamation points (!) operating system commands prefix (isql) 210

executing Adaptive Server 156, 237

See also dataserver utility command 156

See also sqlsrvr utility command 237

### exit command

- in dscp 112
- in isql 10

exporting data. See Copying data out with interactive bcp 76

## F

failure cleanup, **sybmigrate** 304

fast version of bcp 48

- copying data in 50
- data recoverability and 47

### field lengths

- copy in 80
- in interactive bcp 67, 68
- prefix length, integer value in 75

field terminators 61

- bcp 64
- interactive bcp 69, 70

### files

- See also Format files in bcp 72
- default formats with bcp 60
- error (bcp) 87
- native format 60

fixed-length fields 77

format files in bcp 72, 76

- column numbers 73
- column precision 76
- column scale 76
- elements used in 72
- format used for 72

- host file column order 74
- host file data length 76

host file datatype storage formats (SYB) 74

number of records in 73

prefix length, integer value in 75

saving 72

server column names 76

server column orders 76

TDS version number in 73

terminators 76

formatting isql output 10

full path names, specifying in Backup Server 123

## G

GUI mode

for migration 277

## H

help command (dscp) 105, 112

hidden encrypted text 191

**--hide-vcc parameter** 139

high availability

migration with 300

host file

isql and reading 15

host files

column order 74

data length 76

datatype storage formats (SYB) 74

interactive bcp native format 67

## I

IDENTITY columns

bcp and 58

parallel bcp and 58

ignore\_dup\_key option, create index, and bcp 88

image datatype

interactive bcp and 66

importing data. See Copying data in with interactive bcp 79

index re-creation for migration 292

- index threads for **sybmigrate** 270
- indexes
  - bcpl, dropping before using 50
  - slowing down bcpl 47
- insert command
  - bulk copying, comparing 90
- installing
  - character sets (sqllocres) 236
  - character sets, in GUI (sqlloc) 235
  - languages (sqllocres) 236
  - languages, in GUI (sqlloc) 235
  - languages, new (langinstall) 213, 215
  - sort orders (sqllocres) 236
  - sort orders, in GUI (sqlloc) 235
- interactive bcpl 62, 70
  - backslash terminator (\) 69
  - binary data, default 66
  - carriage-return data field terminator (\r) 69
  - character format files, terminators for 69
  - copying data for other software 62
  - copying data in 79, 83
  - copying data out 76, 79
  - defaults for prompts 62, 70
  - field length 67, 68
  - field terminators 69, 70
  - fields, prefix length of 65, 67
  - file storage types 62, 65
  - image datatype, default for 66
  - implicit conversion of datatypes 64, 65
  - null or invisible character terminator (\0) 69
  - null values and 66
  - padding data with spaces 66
  - row terminators 69, 70
  - storage length 67, 68
  - tab data field terminator (\t) 69
  - terminator for new lines (\n) 69
  - terminators for tabular data preparation 69
  - terminators, field and row 68, 70
  - text data copying, default for 66
- interactive SQL parser 201
  - See also isql utility command 201
- interfaces* file
  - for **sybmigrate** 274
- interfaces files
  - backupserver and 126
  - backupserver, specifying names and locations of 122
  - dscpl sessions, opening 106
  - dscpl, viewing and editing with 192
  - dsedit, editing with 94
  - dsedit, viewing and editing in GUI with 193
  - dsedit, viewing and editing with 193
  - opening for editing 92
- iso\_1 character set 83
- isql utility command 201
  - < (redirect in) symbol 15
  - << symbol 209
  - > (redirect out) symbol 15
  - column-separator character 11
  - command terminator 10
  - command terminator, changing 13
  - command terminator, resetting 203
  - correcting typing errors 12
  - defined 201
  - echo input 15
  - examples of 202
  - exit command 10
  - host files, reading 15
  - line numbers, removing 11, 15
  - maximum statement size 10
  - network packet size, setting 15
  - network packet size, specifying in 15
  - output file 11
  - output, formatting 10
  - packet size, setting 15
  - queries, editing 12
  - query buffer, resetting 12
  - quit command in 10
  - reset command in 12
  - statistics 14
  - statistics option with command terminator 14
  - Transact-SQL, using with 10

## J

- Java columns for migration 269
- Java runtime environment migration 265

## L

- LANG environment variable in backupserver 123
- langinstall utility command 213, 215
  - defined 213
  - permissions required for 213
- languages
  - alternate, with bcp 83
  - for migration 269
  - installing and modifying (sqllocres) 236
  - installing and modifying, in GUI (sqlloc) 235
  - installing new (langinstall) 213, 215
  - specification for migration 276
- LC\_ALL environment variable in backupserver 123
- line numbers, removing in isql 11, 15
- load database command and bcp 90
- load transaction command
  - bcp and 90
- loading character sets 153
  - See also charset utility command 153
- locations of backupserver error log files 123
- locking sessions for migration 274

## M

- manual migration, what is not migrated 262
- max memory**, for migration 268
- max network packet size configuration parameter 87
- max network packet size**, configuration for migration 271
- max packet size allowed**, for **sybmigrate** 268
- max parallel degree**, configuration for migration 271
- merging databases, during migration 303
- migrate session, for **sybmigrate** 266
- migrating
  - servers (sybmigrate) 252
- migrating an archive database 266, 272
- modifying
  - character sets (sqllocres) 236
  - character sets, in GUI (sqlloc) 235
  - languages (sqllocres) 236
  - languages, in GUI (sqlloc) 235
  - server entries (dscp) 108
  - server entries with dsedit 97, 98
  - sort orders (sqllocres) 236
  - sort orders, in GUI (sqlloc) 235
- multibyte character sets, for migration 269

## N

- native file format
  - bcp and 60
- native format files 60
- network packet size
  - isql, specifying in 15
- network transport addresses
  - adding with dsedit 99, 101
  - dsedit, editing with 99
- network transports 94
- new servers, creating 246, 247
  - See also srvbuild utility command 246
- new servers, creating, using resource files 248
  - See also srvbuildres utility command 248
- new-line terminator (\\n)
  - in bcp 61
  - in interactive bcp 69
- noncharacter datatypes, operating system format for 65
- non-printable characters, host file 67
- notes
  - ascii\_7 character set compatibility 187
  - batch size settings in bcp 131
  - defncopy and Report Workbench 186
  - hidden encrypted text 191
  - null terminator versus no terminator 70
  - passwords, using 116
  - security 116
  - select into/bulkcopy/pllsort and copying out data in bcp 49
  - slow version of bcp and deadlocks 52
  - special characters, handling in bcp 132
  - system differences in bcp, operating 61
  - triggers not fired by bcp in target table 47
- null character terminator (\\0) in interactive bcp 69
- null values 87
  - interactive bcp and 66
- number (quantity of)
  - server connections to backupserver 122
- number of sort buffers** for migration 271
- number of user connections** for migration 270, 271
- number of worker processes** for migration 271
- numbers
  - line, removing from isql 11, 15
- numeric datatypes
  - bcp conversion to character storage 67

column precision 76  
 column scale 76  
 operating system format for 60

## O

object hierarchy for migration 266  
 objects  
   failed migration of 302  
   re-creating after migration 304  
 opening  
   sessions in dscp 106  
   sessions in dsedit 94  
 operating system files  
   native format 60  
   permission required for copying tables into 44,  
   140  
 operating systems  
   commands prefix (!) (isql) 210  
   non-character datatype formatting 65  
   numeric datatype formatting 60  
**optdiag** command, changes 221  
**optdiag** utility command 216, 223  
   defined 216, 223  
   permissions required for 222  
 output formats, data. See Copying data out with  
   interactive bcp 76

## P

packet size, network  
   bcp, specifying with 86  
   isql, specifying in 15  
   specifying in isql 15  
 padding data and bcp 68  
 parallel bcp  
   copying to a specific partition 53, 57  
   different methods of using 56, 58  
   IDENTITY columns and 58  
   syntax 57  
 parameter  
   **--hide-vcc** 139  
 parentheses ()  
   in SQL statements xv

partitioned tables in bcp 50, 59  
   copying data into randomly 51  
   copying data into, methods for 51  
   page allocations, increasing 53  
 passwords  
   bcp encryption 135  
   defncopy and 186  
   new, generating 160  
   notes about using 116  
 performance  
   bcp issues and 46, 58, 89  
   bulk copy and packet size 87  
   isql network packet size and 15  
 permissions required for  
   backupsrv utility command 127  
   bcp utility command 43, 66  
   charset utility command 153  
   datasrv utility command 161, 241  
   dscp utility command 192  
   dsedit utility command 193  
   dsedit, required for 92  
   langinstall utility command 213  
   operating system file, copying tables into 44, 140  
   optdiag utility command 222  
   sqlloc utility command 235  
   sqllocres utility command 236  
   sqlupgrade utility command 244  
   sqlupgraderes utility command 245  
   srvbuild utility command 247  
   srvbuildres utility command 248  
   tables, copying data into 43, 140  
   xpserver utility command 258  
 post-migration activities, for **sybmigrate** 292  
 prefix field lengths in interactive bcp 65, 67  
 prefix length 75  
 prefix lengths in bcp 64  
 preupgrade utility command 223  
 primary database, restoring after migration 294  
 prompts  
   bcp. See Interactive bcp 62  
 prompts in bcp utility command 63

## Q

query buffer, resetting 12

## Index

- quick reference
    - dscp commands 112
  - quit command
    - in dscp 112
    - in isql 10
  - quotation marks ( ' ) for enclosing special characters 116
  - quotation marks (xd2 xd3 ) for enclosing special characters 116
- ## R
- redirect in symbol (<) in isql 15
  - redirect out symbol (>) in isql 15
  - release path, for **sybmigrate** 265
  - remigrating a database 304
  - remote machines 92
  - remote machines, using dsedit utility on 92
  - renaming server entries with dsedit 97
  - replicate database
    - restoring after migration 295
  - replicated databases
    - post-migration procedures 294
  - Replication Server, migrating replicated data to Adaptive Server 293
  - report Workbench, incompatibility with defncopy 186
  - reports for **sybmigrate** 253, 275
  - reset command (isql) 12
  - resource file mode, for migration 284
  - roll-back processes 84
  - row terminators
    - bcp, in 61
    - interactive bcp, in 69, 70
  - row-level access rules and bcp 84
  - rows in bcp, and erroneous table 88
  - rows, table
    - bulk copying and failed 84
    - length 56
  - rules
    - defncopy, copying with 186
    - tables, copying data into 89
  - errors during copy in operations 87
  - errors during copy out operations 88
  - format files (bcp) 72
  - security Mechanism server attribute (dscp) 108
  - security, notes about 116
  - select a Directory Service screen 92
  - select into command, compared to bulk copying 90
  - select into/bulkcopy/pllsort database option
    - bcp and 49
  - server column
    - names 76
    - orders 76
  - server connections
    - backupserver, number of 122
  - server data, migration of 260
  - server entries
    - adding, with dsedit 96
    - contents, viewing with dscp 111
    - copying, with dsedit 98, 99, 100, 101
    - deleting, with dseditdsedit 97
    - dscp, listing with 110
    - modifying, with dsedit 97, 98
    - renaming, with dsedit 97
    - server entries, adding 98
  - server Entry Editor window 94
  - server Name server attribute (dscp) 107
  - server Object Version server attribute (dscp) 107
  - server Service server attribute (dscp) 107
  - server Status server attribute (dscp) 107
  - servers
    - backupserver, specifying names in 122
    - backupserver, starting in 126
    - migrating 252
    - modes of speed when using bcp 45
    - startserver utility command 249
    - startserver, starting in 249
    - sybmigrate utility command 252
  - sessions
    - listing (dscp) 106
  - setup session, for **sybmigrate** 266
  - server attributes (dscp) 107
  - show servers 229
    - See also showserver utility command 229
  - showserver utility command 229
    - defined 229
    - examples of 229
- ## S
- saving

- size
  - data storage in bcp 64
  - packet size 86
  - text or image data 86
- slow version of bcp 48
  - deadlocks on index pages 52
- sort orders
  - installing and modifying (sqlloccres) 236
  - installing and modifying, in GUI (sqlloc) 235
- sp\_dboption system procedure and bcp 49
- space requirements and bcp steps 50
- special characters
  - bcp, handling in 132
  - utility commands, use in 116
- specifying full path names in Backup Server 123
- specifying server names in backupserver 122
- SPX/IPX addresses for interfaces entries 100
- SQL parser utility. See isql utility command 201
- sql.ini file
  - dsedit sessions, opening with 93
  - entries, adding 96
  - entries, copying to 98, 100, 101
  - entries, deleting 97
  - entries, modifying 96
  - entries, renaming 97
- sqldbgr utility 230
- sqlloc utility command 235
  - defined 235
  - permissions required for 235
- sqlloccres utility command 236
  - defined 236
  - permissions required for 236
- sqlsrvr utility command 237
  - defined 237
- sqlupgrade utility command 227, 244
  - defined 244
  - permissions required for 244
- sqlupgraderes utility command 245
  - defined 245
  - permissions required for 245
- square brackets [ ]
  - in SQL statements xv
- srvbuild utility command 246, 247
  - defined 246
  - permissions required for 247
- srvbuildres utility command 248
  - defined 248
  - permissions required for 248
- starting
  - dscp utility 105
  - dsedit utility command 91, 92
  - servers (startserver) 249
    - XP Server manually (xpserver) 257, 258
- startserver utility command 249
  - backupserver and 126
  - defined 249
  - examples of 249
  - runserver file examples 250
- statistics
  - displaying (optdiag) 216, 223
  - isql 14
  - loading updated (optdiag) 216, 223
- storage types used in bcp 69
- stored procedures
  - copying with defncopy 186
- SYBASE environment variable 265
- SYBASE environment variable in dsedit 93
- SYBASE\_ASE environment variable 265
- SYBASE\_JRE environment variable 265
- sybmigrate**
  - additional network memory** parameter 268
  - character set conversion 268
  - character set specification 276
  - CIS bulk insert array size** 271
  - CIS bulk insert array size** configuration 267
  - CIS bulk insert batch size** configuration 271
  - CIS packet size** configuration 268, 270
  - command line trace flags 276
  - component directory 265
  - components for 263
  - configuration and tuning 269
  - copy threads 270
  - debug level 274
  - dependencies for 263
  - environment variables 265
  - errors to avoid 272
  - executable file 265
  - GUI mode 277
  - help information 274
  - index re-creation 292
  - index threads 270
  - installation 264

- interfaces* file 274
  - Java columns 269
  - Java runtime environment 265
  - language specification 276
  - languages 269
  - lock session override 274
  - max packet size allowed** configuration 268
  - max parallel degree** configuration 271
  - memory for the JVM 265
  - merging two database 303
  - migrate session 266, 282
  - migrating Replication Server data 293
  - migration failure cleanup 304
  - migration of database data 261
  - migration of server data 260
  - migration with high availability 300
  - multibyte character sets 269
  - multiple sessions 267
  - number of sort buffers** configuration 271
  - number of user connections** configuration 270, 271
  - number of worker processes** configuration 271
  - object hierarchy 266
  - objects fail to migrate 302
  - output template resource file 276
  - permissions for 264
  - platforms 265
  - post-migration activities 292
  - post-migration procedures for replicated databases 294
  - pre-migration considerations 267
  - re-creating an object 304
  - release path 265
  - remigrating a database 304
  - reports 253, 275
  - resource file mode 284
  - restoring replicate databases 295
  - restoring the primary database 294
  - setup session 266, 277
  - source Adaptive Server configuration 270
  - starting 273
  - sybmigrate** log 297
  - syntax 274
  - trouble shooting and error messages 302
  - user-defined log file 276
  - validate session 266, 283
  - version string 274
  - sybmigrate .max network packet size** configuration 271
  - sybmigrate utility command 252
    - defined 252
  - sybmigrate,max memory** configuration 268
  - sybmigrate,what is not migrated** 262
  - SYBMIGRATE\_MEMORY environment variable
    - for migration 265
  - sybmultbuf, using to start Backup Server 123
  - symbols
    - in SQL statements xv
  - symbols, field terminator (bcp) 70
  - syntax
    - for **sybmigrate** 274
  - syntax conventions, Transact-SQL xv
  - system data format (SDF) output in bcp 77
  - system procedure (sp\_dboption) 49
- T**
- tab data field terminator (\) in interactive bcp 69
  - table rows
    - copying in batches 84
    - length 56
  - tables
    - bcp character set defaults 134
    - bcp prompts in 63
    - permission required for copying data into 43, 140
    - permission required to copy into operating system files 44, 140
    - utility commands 116
  - tabular data, copying 70
  - tabular output 77, 79
  - TCP/IP addresses for interfaces entries 99
  - terminators (bcp) 76
    - changing 61
    - defined 43
    - field and row, and 68, 70
    - other programs, using for 69
  - text datatype
    - interactive bcp and 66
  - trace flags and backupserver 126
  - trace flags, for **sybmigrate** 276
  - Transact-SQL
    - using with isql 10



transferring data from programs using bcp 42  
 transport Address server attribute (dscp) 107  
 transport Type server attribute (dscp) 107  
 triggers  
   bcp, dropping before using 50  
   copying with defncopy 186  
   slowing down bcp 47  
   tables, copying data into 89  
 trouble shooting, for migration 302  
 troubleshooting dsedit 102

## U

unlogged transactions 47  
 upgrading Adaptive Server 227, 244  
   See also sqlupgrade utility command 244  
 upgrading Adaptive Server using resource files 245  
   See also sqlupgraderes utility command 245  
 utilities, other Adaptive Server, and bcp 89  
 utility commands  
   special characters, using 116  
   table of commands 116

## V

validate session for **sybmigrate** 266  
 viewing interfaces files in GUI 193  
   See also dsedit utility command 193  
 views, copying, with defncopy 186

## W

warnings  
   bcp data loss 49  
   bcp in data recovery 131  
   defncopy failure due to long comments 190

## X

xd2 xd3 (enclosing special characters) 116  
 XP Server, starting manually 257, 258  
   See also xpserver utility command 257

xpserver utility command 257, 258  
   defined 257  
   permissions required for 258

