## Contents

### About This Book ........................................................................................................................... xi

### CHAPTER 1 Building Servers Using dataserver ............................................... 1
   - 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

### CHAPTER 2 Using Interactive isql from the Command Line............................ 9
   - 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

### CHAPTER 3 Using Interactive SQL in Graphics Mode.................................... 17
   - 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
## Contents

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........................................................................ 46  
Bulk copying encrypted data ............................................ 46  
bcp performance ............................................................ 48  
Using fast or slow bcp ...................................................... 49  
Copying in data with fast bcp ............................................ 52  
Bulk copying data into partitioned tables ......................... 53  
Using parallel bulk copy to copy data into a specific partition .. 55  
Using the bcp options..................................................... 62  
Using the default formats ................................................. 62  
Changing terminators from the command line ................. 64  
Changing the defaults: interactive bcp ......................... 64  
Responding to bcp prompts ............................................ 65  
File storage type......................................................... 66
Contents

Prefixed length ............................................................................ 68
Field length.............................................................................. 69
Field and row terminators........................................................ 71
Using format files ..................................................................... 74
Elements of the bcp format file.............................................. 75
Examples: copying out data interactively................................. 79
  Copying out data with field lengths..................................... 79
  Copying out data with delimiters........................................ 80
Examples: copying in data interactively................................. 82
  Copying in data with field lengths...................................... 82
  Copying in data with delimiters.......................................... 84
  Copying in data with a format file....................................... 84
Using bcp with alternate languages....................................... 85
bcp and row-level access rules............................................. 86
Copy in and batch files........................................................... 86
  Improving recoverability..................................................... 87
  Batches and partitioned tables............................................ 88
Copy out and text and image data.......................................... 88
Specifying a network packet size.......................................... 89
Copy in and error files............................................................ 89
Copy out and error files........................................................... 90
Data integrity: defaults, rules, and triggers............................ 91
  Defaults and datatypes....................................................... 91
  Rules and triggers.............................................................. 91
How bcp differs from other utilities..................................... 91

CHAPTER 5  Using dscp ................................................................. 93
  Getting started with dscp.................................................... 93
  Using a dscp session .......................................................... 94
  Working with server entries.............................................. 95
    Adding and modifying server entries.............................. 95
    Copying server entries.................................................. 97
    Listing and viewing contents of server entries.............. 98
    Deleting server entries................................................ 99
  Exiting dscp........................................................................ 100
  Quick reference for dscp utility commands...................... 100

CHAPTER 6  Using dsedit ............................................................. 103
  Getting started with dsedit............................................... 103
  Starting dsedit ................................................................ 103
  Opening an editing session............................................. 105
  Adding, viewing, and editing server entries.................... 107
    Modifying server entries in Windows.......................... 107

Utility Guide
Contents

Modifying server entries in UNIX platforms ........................................ 110
Copying server entries ........................................................................ 112
Troubleshooting dsedit ........................................................................ 114
The dsedit utility does not start ......................................................... 114
Error message: “Unable to open X display” ....................................... 114
Cannot add, modify, or delete server entries .................................... 115

CHAPTER 7 Utility Commands Reference .............................................. 117
Getting started .................................................................................... 118
*_r utilities ....................................................................................... 120
Utilities quick reference .................................................................... 120
Installation or configuration utilities .................................................. 121
Utilities for languages, character sets, and sort orders ..................... 121
Utilities to start servers ..................................................................... 122
Database creation and manipulation utilities ..................................... 122
Utilities to gather information ............................................................ 123
Tuning utility ..................................................................................... 123
Utility to manage a cluster ................................................................. 123
backupserver .................................................................................... 124
cp ...................................................................................................... 130
buildmaster ..................................................................................... 146
certauth ........................................................................................... 147
certpk12 .......................................................................................... 151
certreq ............................................................................................. 154
charset ............................................................................................. 158
cobpre ............................................................................................. 159
cpre ................................................................................................ 160
dataserver ....................................................................................... 161
ddgen ............................................................................................... 168
defncopy .......................................................................................... 192
dscp .................................................................................................. 198
dsedit ............................................................................................... 199
extractjava ........................................................................................ 200
installjava ......................................................................................... 203
isql .................................................................................................... 207
langinstall ......................................................................................... 219
optdiag ............................................................................................. 222
preupgrade ........................................................................................ 229
pwdcrypt ............................................................................................ 234
qptune ............................................................................................... 235
qrmutil ............................................................................................. 242
showserver ....................................................................................... 248
sqldbg ............................................................................................... 249
sqlloc ............................................................................................... 254
Contents

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>show session</td>
<td>318</td>
</tr>
<tr>
<td>show xpserver</td>
<td>320</td>
</tr>
<tr>
<td>shutdown cluster</td>
<td>320</td>
</tr>
<tr>
<td>shutdown instance</td>
<td>321</td>
</tr>
<tr>
<td>start cluster</td>
<td>322</td>
</tr>
<tr>
<td>start instance</td>
<td>322</td>
</tr>
<tr>
<td>upgrade server</td>
<td>323</td>
</tr>
<tr>
<td>use</td>
<td>325</td>
</tr>
</tbody>
</table>

CHAPTER 9  Migration Utility ................................................................. 327

- Overview .......................................................................................... 327
- Benefits of sybmigrate ................................................................. 327
- What sybmigrate does ....................................................................... 328
- What sybmigrate does not do .......................................................... 330
- Before you begin ............................................................................ 331
- Required components for the sybmigrate ......................................... 331
- Dependencies .................................................................................. 331
- Installation .................................................................................... 332
- Permissions .................................................................................... 332
- Platforms ....................................................................................... 333
- Environment settings ...................................................................... 333
- Migrating proxy tables .................................................................... 333
- Migration process ........................................................................... 334
- Overview of the migration process .................................................. 334
- Pre-migration considerations .......................................................... 335
- Configuration and tuning for higher performance .......................... 337
- Possible errors to avoid ................................................................... 340
- Auto-select dependent objects for migration ..................................... 340
- Migrating an archive database ....................................................... 340
- Starting sybmigrate .......................................................................... 341
- GUI mode ....................................................................................... 345
- Resource file mode .......................................................................... 352
- Sybmigrate with Encrypted Columns ................................................ 359
- Post-migration activities ................................................................. 360
- Migrating databases in the Replication Server domain .................... 361
  - Premigration procedures ............................................................. 361
  - Postmigration procedures ........................................................... 362
  - Migrating databases that support wide data ................................. 367
- Limitations ....................................................................................... 368
- Troubleshooting and error messages ............................................... 370
  - Objects fail to migrate .................................................................. 370
  - Beginning database migration ...................................................... 370
  - “Connection refused” and “Unable to obtain connection to the server” ........................................... 370
Contents

Target server cannot be reached from source server ........... 371
If sybmigrate hangs during migration .................................... 371
Merging two databases ......................................................... 371
Post-migration failure cleanup............................................... 372
Remigrating one database .................................................... 372
Re-creating an individual object ............................................ 372
Connection fail....................................................................... 373
“Insufficient memory in JVM shared class”............................ 373
“There is not enough memory in the procedure cache” ...... 373
java.lang related error ........................................................... 373

Index ........................................................................................................................................... 375
About This Book

This manual is a guide to the Sybase® Adaptive Server® Enterprise utility programs available for UNIX and Windows platforms. 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 15.5. 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 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 dscp” – explains how to use the dscp utility to view and edit server entries in the interfaces file in UNIX platforms.

• Chapter 6, “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 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, “Using sybcluster to Manage the Cluster” – describes how to create, start, and administer a shared-disk cluster and the instances in a cluster.

Chapter 9, “Migration Utility” – explains how to use the sybmigrate utility to move data and database schema from pre-12.5 databases into 15.5 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 Adaptive Server® Enterprise documentation set consists of:

- 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. To check for critical product or document information that was added after the release of the product CD, use the Sybase® Product Manuals Web site.

- The installation guide for your platform – describes installation, upgrading, and some configuration procedures for all Adaptive Server and related Sybase products.

- New Feature Summary – describes the new features in Adaptive Server, the system changes added to support those features, and changes that may affect your existing applications.

- Active Messaging Users Guide – describes how to use the Active Messaging feature to capture transactions (data changes) in an Adaptive Server Enterprise database, and deliver them as events to external applications in real time.

- Component Integration Services Users Guide – explains how to use Component Integration Services to connect remote Sybase and non-Sybase databases.

- The Configuration Guide for your platform – provides instructions for performing specific configuration tasks.

- Glossary – defines technical terms used in the Adaptive Server documentation.


- Java in Adaptive Server Enterprise – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.
• *Job Scheduler Users 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).

• *Migration Technology Guide* – describes strategies and tools for migrating to a different version of Adaptive Server.

• *Monitor Client Library Programmers Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.


• *Monitoring Tables Diagram* – illustrates monitor tables and their entity relationships in a poster format. Full-size available only in print version; a compact version is available in PDF format.

• *Performance and Tuning Series* – is a series of books that explain how to tune Adaptive Server for maximum performance:
  
  • *Basics* – contains the basics for understanding and investigating performance questions in Adaptive Server.
  
  • *Improving Performance with Statistical Analysis* – describes how Adaptive Server stores and displays statistics, and how to use the `set statistics` command to analyze server statistics.
  
  • *Locking and Concurrency Control* – describes how to use locking schemes to improve performance, and how to select indexes to minimize concurrency.
  
  • *Monitoring Adaptive Server with sp_sysmon* – discusses how to use `sp_sysmon` to monitor performance.
  
  • *Monitoring Tables* – describes how to query Adaptive Server monitoring tables for statistical and diagnostic information.
  
  • *Physical Database Tuning* – describes how to manage physical data placement, space allocated for data, and the temporary databases.
  
  • *Query Processing and Abstract Plans* – explains how the optimizer processes queries, and how to use abstract plans to change some of the optimizer plans.
  
• *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 (regular size when viewed in PDF format).
• Reference Manual – is a series of books that contains detailed Transact-SQL® information:
  • Building Blocks – discusses datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  • Commands – documents commands.
  • Procedures – describes system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.
  • Tables – discusses system tables, monitor tables, and dbcc tables.
• System Administration Guide –
  • Volume 1 – provides an introduction to the basics of system administration, including a description of configuration parameters, resource issues, character sets, sort orders, and instructions for diagnosing system problems. The second part of Volume 1 is an in-depth discussion about security administration.
  • Volume 2 – includes instructions and guidelines for managing physical resources, mirroring devices, configuring memory and data caches, managing multiprocessor servers and user databases, mounting and unmounting databases, creating and using segments, using the reorg command, and checking database consistency. The second half of Volume 2 describes how to back up and restore system and user databases.
• System Tables Diagram – illustrates system tables and their entity relationships in a poster format. Full-size available only in print version; a compact version is available in PDF format.
• Transact-SQL Users Guide – documents Transact-SQL, the Sybase-enhanced version of the relational database language. This guide serves as a textbook for beginning users of the database management system, and also contains detailed descriptions of the pubs2 and pubs3 sample databases.
• Troubleshooting: Error Messages Advanced Resolutions – contains troubleshooting procedures for problems you may encounter. The problems discussed here are the ones the Sybase Technical Support staff hear about most often.
• Encrypted Columns Users Guide – describes how to configure and use encrypted columns with Adaptive Server.
• In-Memory Database Users Guide – describes how to configure and use in-memory 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 Backup Server with IBM® Tivoli® Storage Manager – describes how to set up and use the IBM Tivoli Storage Manager to create Adaptive Server backups.

• Using Sybase Failover in a High Availability System – provides instructions for using Sybase 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 Users 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 are available in XML services.

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

• The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

• The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.
Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the README.txt file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

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

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

- **Finding the latest information on product certifications**
  2. Click Partner Certification Report.
  3. In the Partner Certification Report filter select a product, platform, and timeframe and then click Go.
  4. Click a Partner Certification Report title to display the report.

- **Finding the latest information on component certifications**
  2. Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
  3. Select Search to display the availability and certification report for the selection.

- **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.
2 Click MySybase and create a MySybase profile.

❖ Finding the latest information on EBFs and software maintenance
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>
<thead>
<tr>
<th>Element</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command names, procedure names, utility names, and other keywords display in sans serif font.</td>
<td><code>select</code> <code>sp_configure</code></td>
</tr>
<tr>
<td>Database names and datatypes are in sans serif font.</td>
<td><code>master database</code></td>
</tr>
<tr>
<td>Element</td>
<td>Example</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| Book names, file names, variables, and path names are in italics. | *System Administration Guide*
| | *sql.ini file*
| | *column_name*
| | *$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. | `select column_name from table_name where search_conditions`
| Type parentheses as part of the command. | `compute row_aggregate (column_name)`
| Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. | `::=`
| Indicates “is defined as”. | `{cash, check, credit}`
| Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces. | `[cash | check | credit]`
| Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets. | `cash, check, credit`
| 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. | `cash | check | credit`
| The pipe or vertical bar (|) means you may select only one of the options shown. | `buy thing = price [cash | check | credit] [, thing = price [cash | check | credit]]...`
| An ellipsis (...) means that you can repeat the last unit as many times as you like. | 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:

```sql
select * from publishers
```

• Examples of output from the computer appear as follows:

<table>
<thead>
<tr>
<th>pub_id</th>
<th>pub_name</th>
<th>city</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0736</td>
<td>New Age Books</td>
<td>Boston</td>
<td>MA</td>
</tr>
<tr>
<td>0877</td>
<td>Binnet &amp; Hardley</td>
<td>Washington</td>
<td>DC</td>
</tr>
<tr>
<td>1389</td>
<td>Algodata Infosystems</td>
<td>Berkeley</td>
<td>CA</td>
</tr>
</tbody>
</table>

(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*.

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.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Building a new master device</td>
<td>2</td>
</tr>
</tbody>
</table>

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

For a detailed description of *dataserver* syntax, see *dataserver* on page 161. For a detailed description of *sqlsrv* syntax, see *sqlsrv* on page 256.

**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 -d device_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 161 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 \((102,400 \times 2K = 200 \text{ MB})\) 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
```

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 10:24:34.74 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 10:24:35.44 server  Database 'master': beginning
upgrade step [ID 20]: creating table (table sysusages)
```

[...]

Adaptive Server Enterprise
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 sybsystemdb 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 sybsystemdb 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.
Building a new master device

Error messages

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

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

```plaintext
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:

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

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

```plaintext
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.

```plaintext
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 9, “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 @@maxpagesize
  -----------
  8192
  ```
Building a new master device
CHAPTER 2

Using Interactive isql from the Command Line

This chapter describes the command line SQL utility, isql.

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

For a detailed description of isql syntax, see isql on page 207.

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 \textit{isql} prompt appears:

\begin{verbatim}
1>
\end{verbatim}

You can now issue Transact-SQL commands.

To exit \textit{isql} enter either of these commands on a line by itself:

\begin{verbatim}
quit
exit
\end{verbatim}

\section*{How to use Transact-SQL in \textit{isql}}

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

\textbf{Note} To use Transact-SQL directly from the operating system with the \textit{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:

\begin{verbatim}
isql
Password:

1> use pubs2
2> go
1> select *
2> from authors
3> where city = "Oakland"
4> go
\end{verbatim}
CHAPTER 2  Using Interactive isql from the Command Line

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:

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

For example, if a column col1 is defined as varchar(10), then `left(col1, 2)` returns a width of eight, or four bytes per character. A value of `left(col1,5)` returns a width or 10, and cannot be larger than the defined length, even though, according to the formula, 5 X 4 equals 20).

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

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

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

```bash
isql -Uuser_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:
  ```bash
type output
  ``
- In UNIX platforms:
  ```bash
cat output
  ```

```bash
select *
```
How to use Transact-SQL in isql

```
from authors
where city = "Oakland"

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

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

<table>
<thead>
<tr>
<th>set option</th>
<th>Default</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>char_convert</td>
<td>Off</td>
<td>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.</td>
</tr>
</tbody>
</table>
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
```

1xact:

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
```

3xact:
Setting the network packet size

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

The `-A` 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_pktsz` 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
CHAPTER 3

Using Interactive SQL in Graphics Mode

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

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

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.

❖ 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.

❖ 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.
Main window description

The top half of the Plan tab shows the logical flow of the operators used in the plan in a tree-based, hierarchal 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 (SDEs, 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>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT+F4</td>
<td>Exits Interactive SQL.</td>
</tr>
<tr>
<td>ALT+LEFT CURSOR</td>
<td>Displays the previous SQL statement in the history list.</td>
</tr>
<tr>
<td>ALT+RIGHT CURSOR</td>
<td>Displays the next SQL statement in the history list.</td>
</tr>
<tr>
<td>CTRL+C</td>
<td>Copies the selected row(s) and column headings to the clipboard.</td>
</tr>
<tr>
<td>CTRL+END</td>
<td>Moves to the bottom of the current pane.</td>
</tr>
<tr>
<td>CTRL+F6</td>
<td>Cycles through the open Interactive SQL windows.</td>
</tr>
<tr>
<td>CTRL+H</td>
<td>Displays the history of your executed SQL statements during the current session.</td>
</tr>
<tr>
<td>CTRL+HOME</td>
<td>Moves to the top of the current pane.</td>
</tr>
</tbody>
</table>
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.

❖ 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.

❖ 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).

❖ 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.

❖ 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.
Looking up tables, columns, and procedures

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

**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.

- **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.

- **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.

- **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.
CHAPTER 3   Using Interactive SQL in Graphics Mode

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.

❖ 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.
CHAPTER 3    Using Interactive SQL in Graphics Mode

- 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:
Configuring Interactive SQL

- 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:

\[
\text{Statement1; OUTPUT TO file1} \\
go \\
\text{statement2; OUTPUT TO file2} \\
go
\]

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

\[
\text{select * from titles} \\
go \\
\text{output to "C:\My Documents\Employees.txt"}
\]

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:

  \[
  \text{read temp.SQL}
  \]

- 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.

❖ 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.

❖ 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.

❖ Running a command file immediately
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:

```sql
{{ 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:

```sql
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**

<table>
<thead>
<tr>
<th>Numeric function names</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
</tr>
<tr>
<td>acos</td>
</tr>
<tr>
<td>asin</td>
</tr>
<tr>
<td>atan</td>
</tr>
<tr>
<td>atan2</td>
</tr>
<tr>
<td>ceiling</td>
</tr>
</tbody>
</table>

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

**Numeric function names**

<table>
<thead>
<tr>
<th>Ascii function names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascii</td>
</tr>
<tr>
<td>char</td>
</tr>
<tr>
<td>concat</td>
</tr>
</tbody>
</table>

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

**System function names**

<table>
<thead>
<tr>
<th>System function names</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
</tr>
</tbody>
</table>
Interactive SQL commands

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))
```

**System function names**

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>curdate</td>
</tr>
<tr>
<td>dayofweek</td>
</tr>
<tr>
<td>monthname</td>
</tr>
<tr>
<td>timestampadd</td>
</tr>
<tr>
<td>curtime</td>
</tr>
<tr>
<td>hour</td>
</tr>
<tr>
<td>now</td>
</tr>
<tr>
<td>timestampdiff</td>
</tr>
<tr>
<td>dayname</td>
</tr>
<tr>
<td>minute</td>
</tr>
<tr>
<td>quarter</td>
</tr>
<tr>
<td>year</td>
</tr>
<tr>
<td>dayofmonth</td>
</tr>
<tr>
<td>month</td>
</tr>
<tr>
<td>second</td>
</tr>
</tbody>
</table>

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

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

For more information about the Interactive SQL commands, see “Using DBISQL Commands” in the Reference Manual: Commands.
CHAPTER 4

Using bcp to Transfer Data to and from Adaptive Server

Versions earlier than 15.0.3 did not allow you to run fast bcp on tables with non-clustered indexes or triggers. Cluster Edition version 15.0.3 and later removes this restriction.

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

---

**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.**
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 130 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:

For a detailed description of bcp syntax, see bcp on page 130.
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`

### Pretransfer 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:

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

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

- To use fast `bcp`, remove indexes on the target table. For more information about this requirement, see “bcp performance” on page 48.

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`:

  ```ini
  [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:

  ```ini
  [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.

Bulk copying encrypted data

bcp transfers encrypted data in and out of databases in either plain text or cipher text form. By default, bcp copies plain text data, processing them as follows:

- Data is automatically encrypted by Adaptive Server before insertion when executing bcp in. Slow bcp is used. The user must have insert and select permission on all columns.

- Data is automatically decrypted by Adaptive Server when executing bcp out. select permission is required on all columns; in addition, decrypt permission is required on the encrypted columns.

This example copies the customer table out as plain text data in native machine format:

```
bcp uksales.dbo.customer out uk_customers -n -Uroy -Proy123
```

If the data to be copied out as plain text is encrypted by a key that uses an explicit password, you can supply that password to bcp using the --c password or --colpasswd options.
For example, if the salary column in the employee table is encrypted by a key that is protected by an explicit password, you can only copy out the salary data as plain text by providing bcp with the password, as follows:

```bash
bcp hr.dbo.employee out  -c -Upjones -PX15tgol --
colpasswd hr.dbo.employee.salary '4mIneIsonly'
```

Alternatively, if you know the name of the key that encrypts the salary column, you can use:

```bash
bcp hr.dbo.employee out  -c -Upjones -PX15tgol --
keypasswd keydb.dbo.hr_key '4mIneIsonly'
```

bcp uses the password to issue a set encryption passwd command before selecting the data.

Use the `--keypasswd` and `--colpasswd` options in a similar way on the bcp command line when copying the data back in.

Use the `-C` option for bcp to copy the data as cipher text. When copying cipher text, you may copy data out and in across different operating systems. If you are copying character data as cipher text, both platforms must support the same character set.

The `-C` option for bcp allows administrators to run bcp when they lack decrypt permission on the data. When the `-C` option is used, bcp processes data as follows:

- Data is assumed to be in cipher text format during execution of bcp in, and Adaptive Server performs no encryption. Use the `-C` option only if the file being copied into Adaptive Server was created using the `-C` option on bcp out. The cipher text must have been copied from a column with exactly the same column attributes and encrypted by the same key as the column into which the data is being copied. Fast bcp is used. The user must have `insert` and `select` permission on the table.

- bcp in `-C` bypasses the domain rule and check constraint for encrypted columns if either exist on an encrypted column because, in this situation, Adaptive Server uses fast bcp. Domain rules and check constraints do not affect bcp out `-C`.

- If an access rule exists on an encrypted column, using bcp out `-C` results in a 2929 error. Access rules do not affect bcp in `-C`.

- Data is copied out of Adaptive Server without decryption on bcp out. The cipher text data is in hexadecimal format. The user must have `select` permission on all columns. For copying cipher text, decrypt is not required on the encrypted columns.
Encrypted char or varchar data retains the character set used by Adaptive Server at the time of encryption. If the data is copied in cipher text format to another server, the character set used on the target server must match that of the encrypted data copied from the source. The character set associated with the data on the source server when it was encrypted is not stored with the encrypted data and is not known or converted on the target server.

You can also perform bcp without the -C option to avoid the character set issue.

You cannot use the -J option (for character set conversion) with the -C option.

The following example copies the customer table. The cc_card column is copied out as human-readable cipher text. Other columns are copied in character format. User “roy” is not required to have decrypt permission on customer cc_card.

```
bcp uksales.dbo.customer out uk_customers -C -c -Uroy -Proy123
```

When copying data as cipher text, ensure that the same keys are available in the database when the data is copied back in. If necessary, use the ddlgen utility to move keys from one database to another.

---

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 55.

**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 uses 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.
bcp performance

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.

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.

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.

Copying tables with indexes
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/pllsort 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/pllsort 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/pllsort option, and whether the transaction log is kept and can be dumped.

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

<table>
<thead>
<tr>
<th>select into/bulkcopy/pllsort</th>
<th>on</th>
<th>off</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast bcp (no cluster indexes on target table)</td>
<td>OK</td>
<td>bcp prohibited</td>
</tr>
<tr>
<td></td>
<td>dump transaction prohibited</td>
<td>dump transaction</td>
</tr>
<tr>
<td>slow bcp (one or more cluster indexes)</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>dump transaction prohibited</td>
<td>dump transaction OK</td>
</tr>
</tbody>
</table>

**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.
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/pllsort` 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/pllsort` 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.

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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Who can do it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use <code>sp_dboption</code> to set <code>select into/bulkcopy/pllsort</code> to true. Run checkpoint in the database that was changed.</td>
<td>System Administrator or Database Owner</td>
</tr>
<tr>
<td>Have enough space to re-create any indexes on the table.</td>
<td>Table owner</td>
</tr>
<tr>
<td>Drop the indexes on the table.</td>
<td></td>
</tr>
<tr>
<td>Have insert permission on the table.</td>
<td>Granted by the table owner</td>
</tr>
<tr>
<td>Perform the copy with bcp.</td>
<td>Any user with insert permission</td>
</tr>
<tr>
<td>Re-create the indexes.</td>
<td>Table owner</td>
</tr>
<tr>
<td>Reset <code>sp_dboption</code>, if desired, and run checkpoint in the database that was changed.</td>
<td>System Administrator or Database Owner</td>
</tr>
<tr>
<td>Use <code>dump database</code> to back up the newly inserted data.</td>
<td>System Administrator, Operator, or Database Owner</td>
</tr>
</tbody>
</table>
CHAPTER 4 Using bcp to Transfer Data to and from Adaptive Server

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 52, 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Who can do it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run stored procedures or queries to determine whether any of the newly loaded data violates rules.</td>
<td>Table owner or stored procedure owner</td>
</tr>
</tbody>
</table>
bcp performance

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

❖ 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 55 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 49 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:
CHAPTER 4   Using bcp to Transfer Data to and from Adaptive Server

- 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:

  
  ```bash
  bcp tablename partition partition_name in file_name
  ```

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

  
  ```bash
  bcp tablename partition_number in file_name
  ```

Figure 4-1 illustrates the parallel bulk copy process.
CHAPTER 4 Using bcp to Transfer Data to and from Adaptive Server

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

Large file divided into four smaller files

File 1
bcp mydb..bigtable:1 in file 2 &

File 2
bcp mydb..bigtable:2 in file 2 &

File 3
bcp mydb..bigtable:3 in file 2 &

File 4
bcp mydb..bigtable:4 in file 2 &

Partitioned table

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:
bcp performance

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

    \[
    \text{# of simultaneous batches} \times \left( \frac{\text{rows per batch}}{2016/\text{row length}} \right)
    \]

  - 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. The smallest batch size bcp can process is 1; the largest is 2147483647L.

- 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 | partition_name] in file_name
-Pmypassw0rd
```

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
- `mypassw0rd` – 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:
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](image)

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

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 \(-g\) 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 [-g id_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.
Using the bcp options

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 130.

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:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0736</td>
<td>New Age Books</td>
<td>Boston</td>
<td>MA</td>
<td></td>
</tr>
<tr>
<td>0877</td>
<td>Binnet &amp; Hardley</td>
<td>Washington</td>
<td>DC</td>
<td></td>
</tr>
<tr>
<td>1389</td>
<td>Algodata Infosystems</td>
<td>Berkeley</td>
<td>CA</td>
<td></td>
</tr>
</tbody>
</table>
Changing the defaults: interactive bcp

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 `-field_terminator` and `-row_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
Changing the defaults: interactive bcp

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

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

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Default provided</th>
<th>Possible user response</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Storage Type</td>
<td>Use database storage type for most fields except:</td>
<td>char to create or read a human-readable file; any Adaptive Server datatype where implicit conversion is supported.</td>
</tr>
<tr>
<td></td>
<td>• char for varchar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• binary for varbinary</td>
<td></td>
</tr>
<tr>
<td>Prefix Length</td>
<td>• 0 for fields defined with char datatype (not storage type) and all fixed-length datatypes</td>
<td>0 if no prefix is desired; otherwise, defaults are recommended.</td>
</tr>
<tr>
<td></td>
<td>• 1 for most other datatypes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 for binary and varbinary saved as char</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 4 for text and image</td>
<td></td>
</tr>
<tr>
<td>Storage Length</td>
<td>For char and varchar, use defined length.</td>
<td>Default values, or greater, are recommended.</td>
</tr>
<tr>
<td></td>
<td>For binary and varbinary saved as char, use double the defined length.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For all other datatypes, use maximum length needed to avoid truncation or data overflow.</td>
<td></td>
</tr>
<tr>
<td>Field or Row Terminator</td>
<td>None</td>
<td>Up to 30 characters, or one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• \t – tab</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• \n – newline</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• \r – carriage return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• \0 – null terminator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• \ – backslash</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Table datatype</th>
<th>Storage type</th>
</tr>
</thead>
<tbody>
<tr>
<td>char, varchar</td>
<td>c[har]</td>
</tr>
<tr>
<td>text</td>
<td>T[ext]</td>
</tr>
<tr>
<td>int</td>
<td>i[n]</td>
</tr>
<tr>
<td>smallint</td>
<td>s[mallint]</td>
</tr>
<tr>
<td>tinyint</td>
<td>t[inyint]</td>
</tr>
<tr>
<td>float</td>
<td>f[loat]</td>
</tr>
<tr>
<td>money</td>
<td>m[oney]</td>
</tr>
<tr>
<td>bit</td>
<td>b[it]</td>
</tr>
<tr>
<td>datetime</td>
<td>d[atetime]</td>
</tr>
<tr>
<td>binary, varbinary, timestamp</td>
<td>x</td>
</tr>
<tr>
<td>image</td>
<td>l[mage]</td>
</tr>
<tr>
<td>smalldatetime</td>
<td>D</td>
</tr>
<tr>
<td>real</td>
<td>r</td>
</tr>
<tr>
<td>smallmoney</td>
<td>M</td>
</tr>
<tr>
<td>numeric</td>
<td>n</td>
</tr>
<tr>
<td>decimal</td>
<td>e</td>
</tr>
</tbody>
</table>

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.
Changing the defaults: interactive bcp

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 78.

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.
Changing the defaults: interactive bcp

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

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Default size</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>12 bytes</td>
</tr>
<tr>
<td>smallint</td>
<td>6 bytes</td>
</tr>
<tr>
<td>tinyint</td>
<td>3 bytes</td>
</tr>
<tr>
<td>float</td>
<td>25 bytes</td>
</tr>
<tr>
<td>money</td>
<td>24 bytes</td>
</tr>
<tr>
<td>bit</td>
<td>1 byte</td>
</tr>
<tr>
<td>datetime</td>
<td>26 bytes</td>
</tr>
<tr>
<td>smalldatetime</td>
<td>26 bytes</td>
</tr>
<tr>
<td>real</td>
<td>25 bytes</td>
</tr>
<tr>
<td>smallmoney</td>
<td>24 bytes</td>
</tr>
</tbody>
</table>

- 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).
CHAPTER 4  Using bcp to Transfer Data to and from Adaptive Server

- 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:
Changing the defaults: interactive bcp

- Tabs, indicated by \t
- New lines, indicated by \n
- Carriage returns, indicated by \r
- Backslash, indicated by \n
- 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.
Using format files

Table 4-6: Adaptive Server char data

<table>
<thead>
<tr>
<th>Prefix length = 0</th>
<th>Prefix length–1, 2, or 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>No terminator</td>
<td>string--string--....</td>
</tr>
<tr>
<td>Terminator</td>
<td>string--Tstring--T...</td>
</tr>
</tbody>
</table>

Table 4-7: Other datatypes converted to char storage

<table>
<thead>
<tr>
<th>Prefix length = 0</th>
<th>Prefix length–1, 2, or 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>No terminator</td>
<td>string--string--....</td>
</tr>
<tr>
<td>Terminator</td>
<td>stringTstringT...</td>
</tr>
</tbody>
</table>

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.
Elements of the \texttt{bcp} format file

The following list names the various elements of a \texttt{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 \texttt{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

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4-3.png}
\caption{\texttt{bcp} format file}
\end{figure}
Using format files

- 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**

<table>
<thead>
<tr>
<th>Storage format</th>
<th>Adaptive Server datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYBCHAR</td>
<td>char, chavarchar (ASCII), nchar, nvarchar</td>
</tr>
<tr>
<td>SYBTEXT</td>
<td>text</td>
</tr>
<tr>
<td>SYBINARY</td>
<td>binary, timestamp, unichar, univarchar, varbinary</td>
</tr>
<tr>
<td>SYBIMAGE</td>
<td>image</td>
</tr>
<tr>
<td>SYBINT1</td>
<td>tinyint</td>
</tr>
<tr>
<td>SYBINT2</td>
<td>smallint</td>
</tr>
<tr>
<td>SYBINT4</td>
<td>int</td>
</tr>
<tr>
<td>SYBINT8</td>
<td>bigint</td>
</tr>
<tr>
<td>SYBFLT8</td>
<td>float</td>
</tr>
<tr>
<td>SYBREAL</td>
<td>real</td>
</tr>
<tr>
<td>SYBBIT</td>
<td>bit</td>
</tr>
<tr>
<td>SYBNUMERIC</td>
<td>numeric</td>
</tr>
<tr>
<td>SYBDecimal</td>
<td>decimal</td>
</tr>
<tr>
<td>SYBMONET</td>
<td>money</td>
</tr>
<tr>
<td>SYBMONET4</td>
<td>smallmoney</td>
</tr>
<tr>
<td>SYBDATETIME</td>
<td>datetime</td>
</tr>
<tr>
<td>SYBDATETIME4</td>
<td>smalldatetime</td>
</tr>
<tr>
<td>SYBDATE</td>
<td>date</td>
</tr>
<tr>
<td>SYBETIME</td>
<td>time</td>
</tr>
<tr>
<td>SYBUNIINT8</td>
<td>unsigned bigint</td>
</tr>
<tr>
<td>SYBUNIINT4</td>
<td>unsigned int</td>
</tr>
<tr>
<td>SYBUNIINT2</td>
<td>unsigned smallint</td>
</tr>
<tr>
<td>SYBUNITEXT</td>
<td>unitext</td>
</tr>
<tr>
<td>SYBFLT8</td>
<td>double</td>
</tr>
</tbody>
</table>

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.

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>
<thead>
<tr>
<th>Length (in bytes)</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No prefix</td>
</tr>
<tr>
<td>1</td>
<td>(2^8-1; 0-255)</td>
</tr>
<tr>
<td>2</td>
<td>(2^{16}-1; 0-65535)</td>
</tr>
<tr>
<td>4</td>
<td>(2^{32}-1; 0-4,294,967,295)</td>
</tr>
</tbody>
</table>

Host file data length

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:

- The maximum field length
- The prefix length, if any
- The field terminator string, if any

If more than one method of field length specification is given, bcp chooses the one that copies the least amount of data.

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 74.

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

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5023</td>
<td>AB-123-DEF-425-1Z3</td>
<td>Oct 31 1985 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>AB-872-DEF-732-2Z1</td>
<td>Nov 6 1985 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>AX-532-FED-452-2Z7</td>
<td>Dec 1 1990 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>BS-345-DSE-860-1F2</td>
<td>Dec 12 1986 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>GH-542-NAD-713-9F9</td>
<td>Mar 15 1987 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>NF-123-ADS-642-9G3</td>
<td>Jul 18 1987 12:00AM</td>
<td></td>
</tr>
<tr>
<td>5023</td>
<td>XS-135-DER-432-8J2</td>
<td>Mar 21 1991 12:00AM</td>
<td></td>
</tr>
</tbody>
</table>
### Examples: copying out data interactively

<table>
<thead>
<tr>
<th>ID</th>
<th>Code</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5023</td>
<td>ZA-000-ASD-324-4D1</td>
<td>Jul 27 1988</td>
<td>12:00AM</td>
</tr>
<tr>
<td>5023</td>
<td>ZD-123-DFG-752-9G8</td>
<td>Mar 21 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>5023</td>
<td>ZS-645-CAT-415-1B2</td>
<td>Mar 21 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>5023</td>
<td>ZZ-999-ZZZ-999-0A0</td>
<td>Mar 21 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>6380</td>
<td>234518</td>
<td>Sep 30 1987</td>
<td>12:00AM</td>
</tr>
<tr>
<td>6380</td>
<td>342157</td>
<td>Dec 13 1985</td>
<td>12:00AM</td>
</tr>
<tr>
<td>6380</td>
<td>356921</td>
<td>Feb 17 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7066</td>
<td>BA27618</td>
<td>Oct 12 1985</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7066</td>
<td>BA52498</td>
<td>Oct 27 1987</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7066</td>
<td>BA71224</td>
<td>Aug 5 1988</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7067</td>
<td>NB-1.142</td>
<td>Jan 2 1987</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7067</td>
<td>NB-3.142</td>
<td>Jun 13 1990</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7131</td>
<td>Asop132</td>
<td>Nov 16 1986</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7131</td>
<td>Asop432</td>
<td>Dec 20 1990</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7131</td>
<td>Fsoap867</td>
<td>Sep 8 1987</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7896</td>
<td>124152</td>
<td>Aug 14 1986</td>
<td>12:00AM</td>
</tr>
<tr>
<td>7896</td>
<td>234518</td>
<td>Feb 14 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>12-F-9</td>
<td>Jul 13 1986</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>13-E-7</td>
<td>May 23 1989</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>13-J-9</td>
<td>Jan 13 1988</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>55-V-7</td>
<td>Mar 20 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>91-A-7</td>
<td>Mar 20 1991</td>
<td>12:00AM</td>
</tr>
<tr>
<td>8042</td>
<td>91-V-7</td>
<td>Mar 20 1991</td>
<td>12:00AM</td>
</tr>
</tbody>
</table>

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 74.

### 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 74.
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 0 4 \"t\" 1 pub_id
2 SYBCHAR 0 40 \"t\" 2 pub_name
3 SYBCHAR 0 20 \"t\" 3 city
4 SYBCHAR 0 2 \"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]: \nDo you want to save this format information in a file? [Y/n] y
Host filename [bcp.fmt]: salesin_fmt
Starting copy...
4 rows copied.
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 Feb 14 1991 12:00AM
6380    342157 Sep 30 1987 12:00AM
6380    356921 Oct 12 1985 12:00AM
7066    BA27618 Aug 14 1986 12:00AM
7066    BA52498 Sep 8 1987 12:00AM
7067    BA71224 Sep 8 1987 12:00AM
7067    NB-1.142 Sep 13 1990 12:00AM
7131    Asoap132 Mar 20 1991 12:00AM
7131    Asoap432 Nov 6 1985 12:00AM
7131    Fsoap867 Feb 14 1991 12:00AM
7896    124152 Aug 14 1986 12:00AM
7896    234518 Nov 6 1985 12:00AM
8042    12-F-9 Jan 13 1988 12:00AM
8042    12-E-7 Oct 12 1985 12:00AM
8042    12-J-9 May 23 1989 12:00AM
8042    13-F-9 Mar 20 1991 12:00AM
8042    13-E-7 Mar 20 1991 12:00AM
8042    13-V-7 Mar 20 1991 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
```

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:
Examples: copying in data interactively

- 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:
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 moresales -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.
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. Reinstate 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 reinstate 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 `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,94705
213-46-8915,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.
213-46-8915 Green Marjorie unknown 309 63rd St. #411
```

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.
**How bcp differs from other utilities**

<table>
<thead>
<tr>
<th>dump database, load database, dump transaction, and load transaction</th>
<th>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 <em>System Administration Guide</em> and the <em>Reference Manual</em>.</th>
</tr>
</thead>
</table>
| 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 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.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started with dscp</td>
<td>93</td>
</tr>
<tr>
<td>Working with server entries</td>
<td>95</td>
</tr>
<tr>
<td>Exiting dscp</td>
<td>100</td>
</tr>
<tr>
<td>Quick reference for dscp utility commands</td>
<td>100</td>
</tr>
</tbody>
</table>

For a detailed description of dscp syntax, see dscp on page 198.

Getting started with dscp

❖ Starting dscp
  • Enter:

        $SYBASE/$SYBASE_OCS/bin/dscp

    The dscp prompt, >>, appears.

❖ 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.

❖ 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>>

❖ Listing all open sessions

- Enter:

  sess

❖ 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.

❖ 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 5-2 describes each attribute.

Table 5-1: Server attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Type of value</th>
<th>Default value and valid values</th>
<th>Can be edited when adding or modifying a server entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Object Version</td>
<td>Integer</td>
<td>110</td>
<td>No</td>
</tr>
<tr>
<td>Server Name</td>
<td>Character string</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Server Service</td>
<td>Character string</td>
<td>SQL SERVER</td>
<td>Yes</td>
</tr>
<tr>
<td>Server Status</td>
<td>Integer</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values are:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 Stopped</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 Failed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 Unknown</td>
<td></td>
</tr>
<tr>
<td>Transport Type</td>
<td>Character string</td>
<td>tcp</td>
<td>Yes</td>
</tr>
<tr>
<td>Transport Address</td>
<td>Character string</td>
<td>None. Valid values are:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decnet, spx, tcp, tli, spx,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tli tcp</td>
<td></td>
</tr>
</tbody>
</table>

Utility Guide 95
Working with server entries

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 5-1 on page 95.

3 To exit add mode, enter:

    #done

Modifying a server entry

You cannot use dscp to modify the Version, Service, and Status entries in the interfaces file.

1 Enter:
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:

```bash
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 5-1 on page 95.

3 To delete an address, enter:
   ```bash
   #del
   ```

4 To exit modify mode, enter:
   ```bash
   #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
Working with server entries

- All entries in the current session to a different session

❖ 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.

❖ 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

❖ 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

❖ 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.
LISTING NAMES OF SERVER ENTRIES

- Enter:
  
  list

LISTING THE ATTRIBUTES OF SERVER ENTRIES

- Enter:
  
  list all

For a description of server attributes, see Table 5-1 on page 95.

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 5-1 on page 95.

DELETING SERVER ENTRIES

You can delete one entry or all entries associated with a session.

DELETING ENTRIES ASSOCIATED WITH A SESSION

- Enter:
  
  del servername

For example, dscp deletes the entry for “myserver” when you enter:

  del myserver
Exiting dscp

- 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 5-2 provides a quick reference to these commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add $servername</code></td>
<td>Adds server entry <code>$servername</code> in the current session. dscp prompts you for information about <code>$servername</code>. Press Return to accept the default value, which is shown in square brackets [ ]. Enter “#done” to exit add mode.</td>
</tr>
<tr>
<td><code>addattr $servername</code></td>
<td>Adds an attribute to the server entry <code>$servername</code> in the current session.</td>
</tr>
<tr>
<td><code>close [sess]</code></td>
<td>Closes a session identified by the <code>sess</code> number. If you do not specify <code>sess</code>, closes the current session.</td>
</tr>
<tr>
<td><code>config</code></td>
<td>Displays configuration information related to your Sybase environment.</td>
</tr>
<tr>
<td>`copy name1 to {name2</td>
<td>sess</td>
</tr>
<tr>
<td></td>
<td>- Server entry <code>name2</code> in the current session,</td>
</tr>
<tr>
<td></td>
<td>- Session <code>sess</code>, or</td>
</tr>
<tr>
<td></td>
<td>- Server entry <code>name2 in session sess</code>.</td>
</tr>
<tr>
<td><code>copyall to sess</code></td>
<td>Copies all server entries in the current session to session <code>sess</code>.</td>
</tr>
<tr>
<td><code>del $servername</code></td>
<td>Deletes server entry <code>$servername</code> in the current session.</td>
</tr>
<tr>
<td><code>delete-all</code></td>
<td>Deletes all server entries in the current session.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits dscp.</td>
</tr>
<tr>
<td><code>help, ?, h</code></td>
<td>Displays the online help.</td>
</tr>
</tbody>
</table>
### CHAPTER 5  Using dscp

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list [all]</td>
<td>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.</td>
</tr>
<tr>
<td>mod servername</td>
<td>Modifies server entry servername in the current session. deep prompts you for information about servername. Press Return to accept the default value, which is shown in square brackets [ ]. Enter “#done” to exit modify mode.</td>
</tr>
<tr>
<td>open [dsname]</td>
<td>Opens a session for the specified directory service, where dsname is the directory service name. If you do not specify a value for dsname, this command opens a session for the default directory service. To open a session, specify the value “InterfacesDriver” for dsname.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits dscp.</td>
</tr>
<tr>
<td>read servername</td>
<td>Displays the contents of server entry servername.</td>
</tr>
<tr>
<td>sess</td>
<td>Lists all open sessions.</td>
</tr>
<tr>
<td>[switch] sess</td>
<td>Makes session number sess the current session.</td>
</tr>
</tbody>
</table>
This chapter explains how to use the dsedit utility to edit the Adaptive Server interfaces file.

### 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 199.

**Note** UNIX users: If your system does not have X-Windows, use dscp to configure server entries in the interfaces file. See Chapter 5, “Using dscp” for more information.

### Starting dsedit

**Windows**

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

- **Starting dsedit from the command prompt**
  - Enter:
    ```
    dsedit
    ```
  - You can specify the following command-line arguments:
Getting started with dsedit

- `-dsname` – 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 `-dsname` argument, `dsedit` presents a list of directory service options in the first dialog box.

- `-l path` – 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`.

Starting `dsedit` through the Windows Explorer

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

Starting `dsedit` from the Sybase for Windows program group

1. Choose Sybase for Windows from the Start menu.

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.

Setting the DISPLAY environment variable

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

   ```
   setenv DISPLAY your_machine_name:0.0
   ```

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

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.

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 112 for more information.

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.
Getting started with dsedit

UNIX platforms

- **Opening the default interfaces file for editing**
  1. Select Sybase Interfaces File.
  2. Click OK.

- **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 107.

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 107.
  • To edit an existing address, click Modify network transport. See “Modifying server entries in Windows” on page 107.
  • 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 6-1.

### Table 6-1: Server attributes

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type of value</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Version</td>
<td>Integer</td>
<td>Version level of the server object definition. Sybase provides this attribute to identify future changes to the object definition.</td>
<td>150</td>
</tr>
<tr>
<td>Server Name</td>
<td>Character string</td>
<td>Server name.</td>
<td>N/A</td>
</tr>
<tr>
<td>Server Service</td>
<td>Character string</td>
<td>A description of the service provided by the server. This value can be any meaningful description.</td>
<td>Adaptive Server</td>
</tr>
</tbody>
</table>
Adding, viewing, and editing server entries

---

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type of value</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Status</td>
<td>Integer</td>
<td>The operating status of the server. Values are:</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Active</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Stopped</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Failed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unknown</td>
<td></td>
</tr>
<tr>
<td>Security Mechanism</td>
<td>Character string</td>
<td>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.</td>
<td>N/A</td>
</tr>
<tr>
<td>Server Address</td>
<td>Character string</td>
<td>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:</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TCP/IP – two formats:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- computer name, port number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ip-address, portnumber</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Named Pipe – pipe name; “pipe” is a required prefix to all pipe names. Server pipes can be only local.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Local – \pipe\sql\query</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Remote – \computer_name\pipe\sql\query</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IPX/SPX – three formats:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- server name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- net number, node number, socket number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- server name, socket number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DECnet – four formats:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- area number, node number, object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- area number, node number, object number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- node name, object name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- node name, object number</td>
<td></td>
</tr>
</tbody>
</table>

---

**Adding a server entry**

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.
❖ **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 6-1 on page 107 for a description of each attribute.
5. Click OK.

❖ **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.

❖ **Deleting a server entry**

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

❖ **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 109 for more information.

❖ **Copying server entries between sessions**

1. Open a session with the directory service or sql.ini file that you want the entries copied to.
Adding, viewing, and editing server entries

2 To open a session, choose File | Open Directory Service. See “Opening additional sessions” on page 105 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 105 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 105 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 105.

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.

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

- Viewing or modifying a server entry
  1 Click on Modify server entry.
  2 Modify the attributes as desired.

- 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 tep-formatted interfaces entries, the host’s IP address and the port number are converted to the 16-byte hexadecimal representation required for tli tep-formatted interfaces entries.

In interfaces entries, use tli tep 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 tep for other clients and servers.
Adding, viewing, and editing server entries

To indicate a TCP/IP address, choose \texttt{tcp} or \texttt{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 \texttt{/dev/mspx} on Solaris and \texttt{/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 \texttt{dsedit}.

To indicate an SPX/IPX address, choose \texttt{tli spx} or \texttt{spx} from the Transport Type menu.

Copying server entries

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

Windows

- \textbf{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.

\texttt{dsedit} appends the copied server entries with a version number of \texttt{_n}. You can rename the copied server entries using Server Object | Rename. See “Renaming a server entry” on page 109 for more information.
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 105 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 94 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 109 for more information.

UNIX platforms

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.
Troubleshooting dsedit
This chapter contains reference pages for the Adaptive Server utility program commands.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started</td>
<td>118</td>
</tr>
<tr>
<td>Utilities quick reference</td>
<td>120</td>
</tr>
<tr>
<td>backupserver</td>
<td>124</td>
</tr>
<tr>
<td>bcp</td>
<td>130</td>
</tr>
<tr>
<td>buildmaster</td>
<td>146</td>
</tr>
<tr>
<td>certauth</td>
<td>147</td>
</tr>
<tr>
<td>certpk12</td>
<td>151</td>
</tr>
<tr>
<td>certreq</td>
<td>154</td>
</tr>
<tr>
<td>charset</td>
<td>158</td>
</tr>
<tr>
<td>cobpre</td>
<td>159</td>
</tr>
<tr>
<td>cpre</td>
<td>160</td>
</tr>
<tr>
<td>dataserver</td>
<td>161</td>
</tr>
<tr>
<td>ddigen</td>
<td>168</td>
</tr>
<tr>
<td>defnscopy</td>
<td>192</td>
</tr>
<tr>
<td>dscp</td>
<td>198</td>
</tr>
<tr>
<td>dsedit</td>
<td>199</td>
</tr>
<tr>
<td>extractjava</td>
<td>200</td>
</tr>
<tr>
<td>installjava</td>
<td>203</td>
</tr>
<tr>
<td>isql</td>
<td>207</td>
</tr>
<tr>
<td>langinstall</td>
<td>219</td>
</tr>
<tr>
<td>optdiag</td>
<td>222</td>
</tr>
<tr>
<td>preupgrade</td>
<td>229</td>
</tr>
<tr>
<td>pwdcrypt</td>
<td>234</td>
</tr>
<tr>
<td>qptune</td>
<td>235</td>
</tr>
<tr>
<td>showserver</td>
<td>248</td>
</tr>
<tr>
<td>sqldbgr</td>
<td>249</td>
</tr>
<tr>
<td>sqlloc</td>
<td>254</td>
</tr>
<tr>
<td>sqllocres</td>
<td>255</td>
</tr>
<tr>
<td>sqlsrvr</td>
<td>256</td>
</tr>
</tbody>
</table>
Getting started

UNIX

You enter a utility program command at the system prompt in a UNIX shell.

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.

Table 7-1: Utility programs

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backupserver</td>
<td>Executable form of the Backup Server™ program.</td>
</tr>
<tr>
<td>bcp</td>
<td>Copies rows in a database table to or from an operating system file in a user-specified format.</td>
</tr>
<tr>
<td>certauth</td>
<td>Converts a server-certificate request into a certificate authority-signed certificate.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>certpk12</td>
<td>Export or import a PKCS#12 file.</td>
</tr>
<tr>
<td>certreq</td>
<td>Creates a server certificate request and corresponding private key in two ways:</td>
</tr>
<tr>
<td>charset</td>
<td>Loads the character sets and sort order files.</td>
</tr>
<tr>
<td>cobpre</td>
<td>Precompiler for COBOL.</td>
</tr>
<tr>
<td>cpre</td>
<td>Precompiler for C.</td>
</tr>
<tr>
<td>dataserver</td>
<td>Executable form of the Adaptive Server program.</td>
</tr>
<tr>
<td>ddlgen</td>
<td>Generates data definition language for server- and database-level objects in ASE.</td>
</tr>
<tr>
<td>defncopy</td>
<td>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.</td>
</tr>
<tr>
<td>dsdp</td>
<td>Allows you to view and edit server entries in the interfaces file in command-line mode.</td>
</tr>
<tr>
<td>dsedit</td>
<td>Allows you to view and edit server entries in the interfaces file using a graphical user interface based on X11/Motif.</td>
</tr>
<tr>
<td>extractjava</td>
<td>Copies a retained JAR from an Adaptive Server to a client file.</td>
</tr>
<tr>
<td>installjava</td>
<td>Installs a JAR from a client file into an Adaptive Server.</td>
</tr>
<tr>
<td>isql</td>
<td>Interactive SQL parser to Adaptive Server.</td>
</tr>
<tr>
<td>langinstall</td>
<td>Installs a new language on the Adaptive Server.</td>
</tr>
<tr>
<td>optdiag</td>
<td>Displays optimizer statistics or loads updated statistics into system tables.</td>
</tr>
<tr>
<td>preupgrade</td>
<td>Performs tests on an installation or database to determine its readiness for upgrade.</td>
</tr>
<tr>
<td>pwdcrypt</td>
<td>Creates and prints an encrypted LDAP password in the libtcl.cfg file.</td>
</tr>
<tr>
<td>qptune</td>
<td>Enables you to fix missing statistics and identify the best query plan, optimization goal, or other configuration settings, and apply them at the query or server level.</td>
</tr>
<tr>
<td>showserver</td>
<td>Shows Adaptive Servers and Backup Servers that are currently running on the local machine.</td>
</tr>
<tr>
<td>sqldbgr</td>
<td>Debugs stored procedures and triggers.</td>
</tr>
<tr>
<td>sqlloc</td>
<td>Installs and modifies languages, character sets, and sort order defaults for Adaptive Server in GUI mode.</td>
</tr>
<tr>
<td>sqllocres</td>
<td>Installs and modifies languages, character sets, and sort order defaults for Adaptive Server in command-line mode.</td>
</tr>
<tr>
<td>sqlsrvr</td>
<td>Executable form of the Adaptive Server program.</td>
</tr>
<tr>
<td>sqlupgrade</td>
<td>Upgrades your currently installed release of Adaptive Server to the newest release in GUI mode.</td>
</tr>
<tr>
<td>sqlupgraderes</td>
<td>Upgrades your currently installed release of Adaptive Server to the newest release in command-line mode.</td>
</tr>
<tr>
<td>srvbuild</td>
<td>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.</td>
</tr>
<tr>
<td>srvbuildres</td>
<td>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.</td>
</tr>
<tr>
<td>startserver</td>
<td>Starts an Adaptive Server or a Backup Server.</td>
</tr>
<tr>
<td>srvmgr</td>
<td>Starts Adaptive Server and Backup Server as Windows services.</td>
</tr>
<tr>
<td>sybcluster</td>
<td>Manages a shared-disk cluster. See Chapter 8, “Using sybcluster to Manage the Cluster.”</td>
</tr>
</tbody>
</table>
Utilities quick reference

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syload</td>
<td>Uploads Sybase products from the distribution media and builds the Sybase installation directory from the command line.</td>
</tr>
<tr>
<td>sybmigrate</td>
<td>Enables you to migrate database from a server using 2K logical pages to a server using 4, 8, or 16K logical pages.</td>
</tr>
<tr>
<td>sybsetup</td>
<td>Installs and configures Adaptive Server from a single location using a GUI interface.</td>
</tr>
<tr>
<td>xpserver</td>
<td>Starts XP Server manually.</td>
</tr>
</tbody>
</table>

* _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 121
- “Utilities for languages, character sets, and sort orders” on page 121
- “Utilities to start servers” on page 122
- “Database creation and manipulation utilities” on page 122
- “Utilities to gather information” on page 123
- “Tuning utility” on page 123
- “Utility to manage a cluster” on page 123
CHAPTER 7  Utility Commands Reference

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 quick reference

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.

- **sqlsrvr**: 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.

- **ddlgen**:Generates data definition language for server- and database-level objects in ASE.

- **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.

Tuning utility

Use the following utility to tune the Adaptive Server:

- **qptune**: Enables you to fix missing statistics and identify the best query plan, optimization goal, or other configuration settings, and apply them at the query or server level.

Utility to manage a cluster

Use the following utility to manage an Adaptive Server shared-disk cluster:

- **sybcluster**: Provides a set of interactive, command line options for creating and managing a cluster. See Chapter 8, “Using sybcluster to Manage the Cluster.”
backupserver

Description
The executable form of the Backup Server program, located in $SYBASE/$SYBASE_ASE/bin.

Windows
The utility is bcksvr.exe, located in %SYBASE%\%SYBASE_ASE%\bin.

Syntax
backupserver

[-C server_connections]
[-S b_servername]
[-I interfaces_file]
[-e error_log_file]
[-M symmultbuf_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]

Or
backupserver -v

Parameters
-C server_connections
specifies the number of server connections for the Backup Server. The Backup Server requires:

• Two connections for each dump session
• One connection for each load session
• One connection for volume change messages

Allow a maximum of three times the number of expected concurrent dump and load sessions. The default value is 30 server connections.

-S b_servername
specifies the name of the Backup Server to start. The default is SYB_BACKUP. This entry must specify the name of a Backup Server in the interfaces file.
-I interfaces_file
  specifies the name and location of the interfaces file to search when
correcting to Backup Server. If -I is omitted, backupear 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 backupear
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:
- -V4 – displays all -V0 messages except “Connection from Server”
  messages printed for each connection event.
- -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
backupserver

-\(n\)
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 DLISTEN, `backupserver` uses the default Backup Server name SYB_BACKUP in UNIX.

  **In Windows** – `bcksrvr` uses the default Backup Server name `server_name_BS`. The value of the DLISTEN environment variable overrides this default value, and the \(-S\) parameter overrides both the default and the value specified in DLISTEN.

- 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
CHAPTER 7 Utility Commands Reference

- 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
[-f formatfile]
[-e errfile]
[-d discardfileprefix]
[-F firstrow]
[-L lastrow]
[-b batchsize]
[-n]
[-c]
[-t field_terminator]
[-r row_terminator]
[-U username]
[-P password]
[-I 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]
[-W]
[-X]
[-M LabelName LabelValue]
[-labeled]
[-K keytab_file]
[-R remote_server_principal]
[-C]
[-V {security_options}]
[-Z security_mechanism]
[-Q]
[-Y]
[-y sybase directory]
[-x trusted.txt_file]
[--maxconn maximum_connections]
[--show-fi]
[--hide-vcc]
[--colpasswd [[[database_name.][owner.]}table_name.]column_name}
[password]
[-keypasswd [(database_name,[owner],)key_name [password]]]

Or

bcp -v

Parameters

maxconn  maximum_connections
is the maximum number of parallel connections permitted for each bulk
copy operation. The default is 10.

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.

-d discardfileprefix
    Logs the rejected rows into a dedicated discard file. The discard file has the
    same format as the host file and is created by appending the input file name
    to the discard file prefix supplied. You can correct the rows in this file and
    use the file to reload the corrected rows.

    Sybase recommends that you use the -d discardfileprefix option in
    conjunction with -e errfile to help identify and diagnose the problem rows
    logged in the discard file.

-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). If multiple files are used, this option applies to each file.
**-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. The largest number bcp accepts for batchsize is 2147483647L.

**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 -Ppassword, 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.

-W
specifies that if the server to which bcp is attempting to connect supports neither normal password encryption nor extended password encryption, plain text password retries are disabled. If this option is used, the CS_SEC_NON_ENCRYPTION_RETRY connection property will be set to CS_FALSE, and plain text (unencrypted) passwords will not be used in retrying the connection.

-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.
-M LabelName LabelValue
   (secure Adaptive Server only) enables multilevel users to set the session
   labels for the bulk copy. Value values for LabelName are:
   
   - currread (current reading level) – is the initial level of data that you can
     read during this session, currread must dominate curwrite.
   
   - curwrite (current write level) – is the initial sensitivity level that will be
     applied to any data that you write during this session.
   
   - maxread (maximum read level) – is the maximum level at which you
     can read data. This is the upper bound to which you as a multilevel user
     can set your currread during the session. maxread must dominate
     maxwrite.
   
   - maxwrite (maximum write level) – is the maximum level at which you
     can write data. This is the upper bound to which you as a multilevel user
     can set your curwrite during a session. maxwrite must dominate minwrite
     and curwrite.
   
   - minwrite (minimum write level) – is the minimum level at which you
     can write data. This is the lower bound to which you as a multilevel user
     can set curwrite during a session. minwrite must be dominated by
     maxwrite and curwrite.

   LabelValue is the actual value of the label, expressed in the human-readable
   format used on your system (for example, “Company Confidential
   Personnel”).

-labeled
   (secure Adaptive Server only) indicates that the data you are importing
   already has labels in the first field of every record.

   For exporting data, this option indicates that you want the sensitivity label
   of every row to be copied out as the first field.

-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.
-C
supports bulk copy of encrypted columns if Adaptive Server supports encrypted columns. -C enables the ciphertext option before initiating the bulk copy operation.

-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

-i input_file
specifies the name of the input file. The Standard Input is used as the default.

-o output_file
specifies the name of the output file. The Standard Output is used as the default.

-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.

-`y sybase_directory`
  sets an alternate Sybase home directory.

-`x trusted.txt_file`
  specifies an alternate trusted.txt file.

--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.

--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.

--colpasswd [[database_name [owner].]table_name.]column_name [password]]
  sets the passwords for encrypted columns by sending set encryption passwd password for column column_name to Adaptive Server. This does not automatically apply passwords to other encrypted columns, even if the second column is encrypted with the same key. The password must be supplied a second time to access the second column.

--keypasswd [[database_name.[owner].]key_name [password]]
  sets passwords for all columns accessed by a key by sending set encryption passwd password for key key_name to Adaptive Server.
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 -c -t , -r \r
```
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\a, \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.
- You can use bcp to copy encrypted data in and out of the server. See “Bulk copying encrypted data” on page 46.
- 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-vcc improves performance because 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

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>bcp</td>
<td>bcp in</td>
<td>• Roles – Current active roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Keywords or options – NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Previous value – NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Current value – NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Other information – NULL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Proxy information – Original login name, if set proxy in effect</td>
</tr>
</tbody>
</table>

| 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 161 for syntax.

**Syntax**
None.
certauth

Description

Note OpenSSL is distributed as a convenience in the Open Client Server (OCS) release. OpenSSL is documented at http://www.openssl.org

Converts a server certificate request to a CA- (certificate authority) signed certificate. Located in $SYBASE/$SYBASE_OCS/bin.

Windows The utility is certauth.exe, and is located in %SYBASE%\%SYBASE_OCS%\bin.

Syntax

certauth

[-r]
[-C caCert_file]
[-Q request_filename]
[-K caKey_filename]
[-N serial_number]
[-O SignedCert_filename]
[-P caPassword]
[-s start_time]
[-T valid_time]

Or

certauth -v

Parameters

- r
  when specified, creates a self-signed root certificate for the test environment.

- C caCert_file
  specifies the name of the CA's certificate request file when -r is specified, or specifies the name of the CA's root certificate.

- Q request_filename
  specifies the name of certificate request file.

- K caKey_filename
  specifies the name of the CA's private key.

- N serial_number
  specifies the serial number in the signed certificate. If -N is not specified, certauth generates a pseudo-random serial number.

- O SignedCert_filename
  specifies the name to use for the output when creating a signed certificate file. If -r is specified, SignedCert_filename is the self-signed root certificate. If -r option is not used, SignedCert_filename is the certificate signed by the caCert_file.
-P caPassword
specifies the CA’s password that is used to decrypt its private key.

-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 SUCCEED (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-----
MIICSTCCAgUCAwAwH3g2IzgEAwUBAAMG8x+CzAJBgNVBAYTAVMBIjJGSDBDQYJKoZIzj0IMA0G
+A1UEAxQaYXlcYXNlMQswCgYDVQQLFANEU1QxFzAVBgNVBAMUDnN5YmFzZV90ZXN0X2Nh
Bgcqkhjkj0OAQBDMGcAkEA+6xG7XCi1xbP96nHBnQtTLC7hc1cyQ8h1ekwv901gG
EMG9AhjLXj6VckPQD75vqVMEkapaPjIoIBHEjE/aXYQIVApyvY1+B9phCa2e2Yf7
cReCsnAxkB77rnOJZLnd8iLQ0t0wd1w4lo/Xx2OeS4CJW0KVkkGd1hNGZ8r
GrQTspWcThrNGBx1XNhAV5g40CgrYAmAAk70uNEI90KmhdR3IsiceCMgQf
1J8dgtWF15cmHeS80mF9s/vqPfrSNkaVh7LJk6kk7VxUXBy+8LMUgpJF/TYMAgsG
BygSM44BAMFAAMx2DAtUAhU2Icn1pSavQtXFzXJUCoOmNLpCFQDte8RUGuo8
ZdxnQtp9uJDMBiUQ==
-----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.

- Adaptive Server includes the openssl open source utility in $SYBASE/$SYBASE_OCS/bin. Use openssl to accomplish all certificate management tasks implemented by certreq, certauth and certpk12. Sybase includes this binary as a convenience, and is not responsible for any issues incurred using the binary. See at www.openssl.org for details.

- 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

Note OpenSSL is distributed as a convenience in the Open Client Server (OCS) release. Documentation is available at the OpenSSL Web site at http://www.openssl.org

Export or import a PKCS #12 file into a certificates file and a private key. Located in $SYBASE/$SYBASE_OCS/bin.

Windows The utility is certpk12.exe, and is located in %SYBASE%%SYBASE_OCS%\bin.

Syntax

certpk12

{ -O Pkcs12_file | -I Pkcs12_file } 
[-C Cert_file] 
[-K Key_file] 
[-P key_password] 
[-E Pkcs12_password]

Or

certpk12 -v

Parameters

-O Pkcs12_file
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 -O or -I needs to be on.

-I Pkcs12_file
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 -I or -O needs to be on.

-C Cert_file
specifies the name of certificate file to be exported to a PKCS #12 file if -O is on; or the name of certificate file to be imported from a PKCS #12 file if -I is on.

-K Key_file
specifies the name of private key file to be exported to a PKCS #12 file if -O is on; or the name of private key file to be imported from a PKCS #12 file if -I is on.

-P Key_password
specifies the password which is used to protect the private key specified by -K. If -O is on, the password is required to export the private key to a PKCS #12 file; if -I is on, the password is required to output the private key to a text file after it is imported from a PKCS #12 file.
-E  

 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
```

**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
```

**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
```

**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
```

---

Adaptive Server Enterprise
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.
- Adaptive Server includes the openssl open source utility in $SYBASE/$SYBASE_OCS/bin. Use openssl to accomplish all certificate management tasks implemented by certreq, certauth and certpk12. Sybase includes this binary as a convenience, and is not responsible for any issues incurred using the binary.

  See the Open Source SSL website at [www.openssl.org](http://www.openssl.org) for details.

- 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.

Note

OpenSSL is distributed as a convenience in the Open Client Server (OCS) release. Documentation is available at the OpenSSL Web site at http://www.openssl.org.

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.

- Adaptive Server includes the openssl open source utility in $SYBASE/$SYBASE_OCS/bin. Use openssl to accomplish all certificate management tasks implemented by certreq, certauth and certpk12. Sybase includes this binary as a convenience, and is not responsible for any issues incurred using the binary. See the Open Source SSL website at www.openssl.org for details.

- 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**

**UNIX platforms only** Loads the character sets and sort order files in Adaptive Server. Located in $SYBASE/$SYBASE_ASE/bin.

**Syntax**

```
charset
    [-P password]
    [-S server]
    [-I interface]
    sort_order
    [ charset ]
```

Or

```
charset -v
```

**Parameters**

- `-P password` specifies your password. If you do not specify `-P`, `charset` prompts for your password.

- `-S server` specifies the name of the server on which to change the character set and sort order.

- `-I interface` specifies the network interface used by the server.

- `sort_order` specifies the name of the sort order file Adaptive Server will use.

- `charset` specifies the character set Adaptive Server will use.

- `-v` displays the version number and copyright message for `charset`.

**Usage**

- Before using `charset`, you must set your SYBASE environment variable to point to the current release directory.

**Permissions**

You must be a System Administrator to use `charset`.

**See also**

- **Commands** `set`
- **Utilities** `langinstall`
cobpre

Description
Precompiler for COBOL, located in $SYBASE/$SYBASE_OCS/bin (%SYBASE%\%SYBASE_OCS%\bin in Windows). For a full description of cpre, see Appendix A of the Open Client and Open Server Programmer’s Supplement.

Syntax
See above.
<table>
<thead>
<tr>
<th><strong>cpre</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
</tr>
</tbody>
</table>
dataserver

**Description**

*UNIX platforms only* The executable form of the Adaptive Server program, located in `$SYBASE/$SYBASE_ASE/bin`.

**Syntax**

dataserver

- `-u`, `--admin-name=sa/sso_name`  
  `--buildquorum={force}`
- `-a`, `--caps-file=filename`
- `-F`, `--cluster-input=filename`
  `--cluster-takeover`
- `-L`, `--conn-config-file=filename`
  `--create-cluster-id [=quorum]`
- `-D`, `--default-db-size=size_spec`
- `-e`, `--error-log=filename`
- `-G`, `--event-log-server=logserv_name`
  `--forcebuild`
- `-H`, `--ha-server`
- `-h`, `--help=[[0|1|2|3],display_width]]`
  `--instance=instance_name`
- `-y`, `--key-password=[key_password]`
- `-K`, `--keytab-file=filename`
- `-N`, `--license-prop-file=filename`
- `-a`, `--logical-page-size=page_size`
- `-Z`, `--master-db-size=size_spec`
- `-d`, `--master-dev=master_device_name`
- `-b`, `--master-dev-size=[size_spec]`
- `-r`, `--master-mirror=filename`
- `-m`, `--masterrecover`
- `-g`, `--no-event-logging`
- `-Q`, `--quorum-dev=quorum_dev`
- `-q`, `--recover-quieted`
- `-w`, `--rewrite-db=database_name`
- `-p`, `--sa-name={SSO_login_account | sso_role | sa_role}`
- `-k`, `--server-principal=s_principal`
- `-M`, `--shared-mem-dir=directory_name`
- `-X`, `--sybmon`
- `-T`, `--trace=trace_flag`
- `-v`, `--version`

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 166 and “Dependencies and conditions of `-b` and `-w` options” on page 166 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.

-k, --server-principal=s_principal
specifies the server principal name.

-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>.crt.

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 Specifies the “aseprincipal@MYREALM.COM” principal name:

```
$SYBASE/$SYBASE_ASE/bin/dataserver -d master.dat -s secure_ase -k aseprincipal@MYREALM.COM
```

Example 4 Rewrites a corrupt master database, specifying device size:

```
dataserver -d my_master_device -w master -z 4k
```

Example 5 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 6 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
```

Example 7 Rewrites a corrupt master database, specifying an incorrect page size, even in a normal restart. This produces a failure:

```
dataserver -d my_master_device -z 4000
```

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_8 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 sybsystemprocs 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**

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
   -U login
   -P password
   -S[[ssl:]server | host_name : port_number]
   [-I interfaces_file]
   [-Tobject_type]
   [-Nobject_name]
   [-Ddbname]
   [-Xextended_object_type]
   [-Ooutput_file]
   [-Eerror_file]
   [-Lprogress_log_file]
   [-Jclient_charset]
   [-LC -N logical_cluster_name]
   [-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 187.
CHAPTER 7 Utility Commands Reference

-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] -I – ddlgen parses the interfaces file specified at the user location for the server name (see the -I 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.

- -I
  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:

<table>
<thead>
<tr>
<th>Object type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>cache</td>
</tr>
<tr>
<td>D</td>
<td>default</td>
</tr>
<tr>
<td>DB</td>
<td>database</td>
</tr>
<tr>
<td>DBD</td>
<td>database device</td>
</tr>
<tr>
<td>Object type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DPD</td>
<td>dump device</td>
</tr>
<tr>
<td>EC</td>
<td>execution class</td>
</tr>
<tr>
<td>EG</td>
<td>engine group</td>
</tr>
<tr>
<td>EK</td>
<td>encrypted keys</td>
</tr>
<tr>
<td></td>
<td>For more information on encrypted keys, see “Using ddlgen for encrypted columns” on page 187</td>
</tr>
<tr>
<td>GRP</td>
<td>group</td>
</tr>
<tr>
<td>I</td>
<td>index</td>
</tr>
<tr>
<td>KC</td>
<td>key constraints</td>
</tr>
<tr>
<td>L</td>
<td>login</td>
</tr>
<tr>
<td>LK</td>
<td>logical key</td>
</tr>
<tr>
<td>P</td>
<td>stored procedure</td>
</tr>
<tr>
<td>R</td>
<td>rule</td>
</tr>
<tr>
<td>RI</td>
<td>referential integrity</td>
</tr>
<tr>
<td>RO</td>
<td>role</td>
</tr>
<tr>
<td>RS</td>
<td>remote server</td>
</tr>
<tr>
<td>SGM</td>
<td>segment</td>
</tr>
<tr>
<td>TR</td>
<td>trigger</td>
</tr>
<tr>
<td>U</td>
<td>table</td>
</tr>
<tr>
<td>UDD</td>
<td>user-defined datatype</td>
</tr>
<tr>
<td>USR</td>
<td>user</td>
</tr>
<tr>
<td>V</td>
<td>view</td>
</tr>
<tr>
<td>WS</td>
<td>user-defined Web service</td>
</tr>
<tr>
<td>WSC</td>
<td>Web service consumer</td>
</tr>
<tr>
<td>XOD</td>
<td>Local caches</td>
</tr>
<tr>
<td>XOU</td>
<td>Global caches</td>
</tr>
<tr>
<td>XP</td>
<td>extended stored procedure</td>
</tr>
</tbody>
</table>
-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 an encrypted key with -Ndb_name.owner.key_name.
- 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, \texttt{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.
- **in-memory databases, caches, and devices** – use the OI extended object type with the -X option.
- **in-memory temporary databases** – use the OIT extended object type with the -X option.

\textbf{Note} \texttt{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.

-Eerror_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 \texttt{ddlgen}. If you do not specify -L, the progress is not recorded.
-Jclient_charset

specifies the character set to use on the client. -Jclient_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.

-LC

generate DDL for one or all logical clusters on a server.
-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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Filters out</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>Everything, and retrieves the schema-only definition of a database or table.</td>
</tr>
<tr>
<td>SGM</td>
<td>Segments</td>
</tr>
<tr>
<td>GRP</td>
<td>Groups</td>
</tr>
<tr>
<td>USR</td>
<td>Users</td>
</tr>
<tr>
<td>R</td>
<td>Rules</td>
</tr>
<tr>
<td>D</td>
<td>Defaults</td>
</tr>
<tr>
<td>UDD</td>
<td>User-defined datatypes</td>
</tr>
<tr>
<td>U</td>
<td>User tables</td>
</tr>
<tr>
<td>V</td>
<td>Views</td>
</tr>
<tr>
<td>P</td>
<td>Stored procedures</td>
</tr>
<tr>
<td>PC</td>
<td>Partition condition</td>
</tr>
<tr>
<td>XP</td>
<td>Extended stored procedures</td>
</tr>
<tr>
<td>I</td>
<td>Indexes</td>
</tr>
<tr>
<td>RI</td>
<td>Referential integrity constraints</td>
</tr>
<tr>
<td>KC</td>
<td>Primary- and unique-key constraints</td>
</tr>
<tr>
<td>TR</td>
<td>Triggers</td>
</tr>
</tbody>
</table>

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.
**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.phonedflt -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 -TD -N dbname.owner.defaultname
```

To generate DDL for all defaults in a database owned by “owner”:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TD -Nowner.% -D dbname
```

**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 -Nmastert
```

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:
To generate DDL for all temporary databases, use the OD extended database type:

```
ddlgen -U login -P password -S server:port -TDB -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 -U login -P password -S server:port -TDB -XOU -N%
```

The following generates DDL for a temporary database named `tempdb1`:

```
ddlgen -U login -P password -S server:port -TDB -N tempdb1
```

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** In-memory databases – generates DDL for an in-memory database:

```
ddlgen -U roy -P roy123 -S SHARBOR:1955 -TDB -N imdb_1
```

**Example 7** Generates DDL for an in-memory temporary database:

```
ddlgen -U roy -P roy123 -S SHARBOR:1955 -TDB -N imdb_temp1
```
**Example 8** 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 9** 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 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 in the authors database on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEK -Nauthors.dbo.%
```

**Example 10** Generates DDL for all encryption keys in the accounts database on a machine named “HARBOR” using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TEK -Naccounts.dbo.%
```

Alternatively, you use the -D option to specify the database name.

**Example 11** Generate DDL for an encryption key “ssn_key” in a the SampleKeysDB database:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key
```

**Example 12** 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
```

**Example 13** Generates DDL for “ssn_key” and its key copies:

```
ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key
```

**Example 14** This 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 -N db1.dbo.employee
```

**Example 15** Engine groups – generates DDL for an engine group called LASTONLINE running on a machine named HARBOR using port 1955:
To generate DDL for all engine groups:

```bash
ddlgen -U roy -P proy123 -S SHARBOR:1955 -TEG -NLASTONLINE
```

**Example 16** Execution class – generates DDL for an execution class called EC2 running on a machine named HARBOR using port 1955:

```bash
ddlgen -U roy -P proy123 -S SHARBOR:1955 -TEC -NEC2
```

To generate DDL for all execution classes:

```bash
ddlgen -U login -P password -S server:port -TEC -N%
```

**Example 17** 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:

```bash
ddlgen -U roy -P proy123 -S SHARBOR:1955 -TXP -N pubs2.jones.xp_cmdshell
```

Alternatively, you can use the -D option instead of using the fully qualified name:

```bash
ddlgen -U login -P password -S server:port -TXP -Nowner.extendedstoredprocedure -D dbname
```

To generate DDL for all extended stored procedures:

```bash
ddlgen -U login -P password -S server:port -TXP -N dbname.
```

**Example 18** 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:

```bash
ddlgen -U roy -P proy123 -TU -N authors -D pubs2 -F%
```

Alternatively, you can specify each of the filters individually:

```bash
ddlgen -U login -P password -TU -N dbname.owner.table -FI,RI,KC,TR
```

The following generates the definition of `table_name` while filtering out foreign keys and primary-unique keys:

```bash
ddlgen -U login -P password -TU -N table_name -D dbname -FRI,KC
```

Both of these generate foreign keys for a specified user in the entire database:

```bash
ddlgen -U login -P password -TRI -N%.%.% -D dbname
```
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% -D dbname
```

The following generates schema-only definition of a database:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TF -N dbname -F%
```

Alternatively, you can specify each of the filters individually:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -N dbname -FSGM,GRP,USR,R,D,UDD,V,XP,I,RI,KC,TR
```

The following generates the database DDL skipping the compiled object:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -N dbname -FTR,D,XP,V,R
```

The following generates database definition without a table definition:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TDB -N dbname -FU
```

**Example 19 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 -TDB -N dbname -D
```

To generate DDL for all groups:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TGRP -N dbname.
```

**Example 20 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:
If you use a fully qualified name, you may omit the -D option.

To generate DDL for all indexes for a single table:
```
ddlgen -U login -P password -S server:port -TI -N dbname.owner.tablename.indexname
```

To generate DDL for all indexes of all tables in a database:
```
ddlgen -U login -P password -S server:port -TI -N dbname.%.%.%
```

For example, this generates DDL for all indexes for all tables in the pubs2 database:
```
ddlgen -Usa -P -SHARBOR:1955 -TI -Npubs2.%.%.%
```

Example 21 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 -N dbname.%.%.PK%
```

Or:
```
ddlgen -Usa -P -TKC -N dbname.%.%.PK% -D dbname
```

Example 22 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 23 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 `-Username` instead of `-N%

```
ddlgen -U login -P password -S server:port -TL -Username
```

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 -U roy -P roy123 -S HARBOR:1955 -TL -N george
```

### Example 24 Remote Servers
Generates DDL for a remote server called ORANGE on a machine named HARBOR using port 1955:

```
ddlgen -U roy -P roy123 -S HARBOR:1955 -TRS -N ORANGE
```

To generate DDL for all remote servers:

```
ddlgen -U login -P password -S server:port -TRS -N%
```

### Example 25 Roles
Generates DDL for the sa_role on a machine named HARBOR using port 1955:

```
ddlgen -U roy -P roy123 -S HARBOR:1955 -TRO -N sa_role
```

To generate DDL for all roles:

```
ddlgen -U login -P password -S server:port -TRO -N%
```

**Note** The password in the DDL generated for all roles is “password”.

### Example 26 Rules
Generates DDL for all rules associated with authors on a machine named HARBOR using port 1955:

```
ddlgen -U roy -P roy123 -S HARBOR:1955 -TR -N authors.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 -U login -P password -S server:port -TR -N dbname.owner.rulename
```

Alternatively, you can also use the `-D` parameter:

```
ddlgen -U login -P password -S server:port -TR -D owner.rulename -D dbname
```

### Example 27 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 -U roy -P roy123 -S HARBOR:1955 -TSGM -N pubs2.logsegment
```
Alternatively, you can use specify the dbname using the -D option:

```
ddlgen -U login -P password -S server:port -TSGM -N segmentname -D dbname
```

To generate DDL for all segments:

```
ddlgen -U login -P password -S server:port -TSGM -N dbname.%
```

**Example 28 SQLJ functions** – Generates DDL for a SQLJ function named region_of owned by dbo in database master:

```
ddlgen -U roy -P roy123 -S SHARBOR:1955 -T -N master.dbo.region_of
```

Alternatively you can also use the -D parameter:

```
ddlgen -U login -P password -S server:port -T -N dbo.region_of -D master
```

To generate DDL for all SQLJ functions in a database, use object type F:

```
ddlgen -U login -P password -S server:port -T -N dbname.owner.%
```

**Example 29 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 -U roy -P roy123 -S SHARBOR:1955 -T -N master.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 -U login -P password -S server:port -T -N master.dbo.% -X OD
```

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 -U login -P password -S server:port -T -N master.dbo.% -X OU
```

**Example 30 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 -U roy -P roy123 -S SHARBOR:1955 -T -N pubs2.dbo.sp_monitor
```

Alternatively, you can use specify the dbname using the -D option:

```
ddlgen -U login -P password -S server:port -T -N owner.procedurename -D dbname
```

To generate DDL for all stored procedures:

```
ddlgen -U login -P password -S server:port -T -N dbname.owner.%
```
Example 31 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 32 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 33 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:

```
-NDdbname.table_owner.table_name.trigger_name
```

To generate DDL for all triggers of a database:
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 34 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 -Nname.%
```

**Example 35 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.%
```

**Example 36 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 -Shost_name:port -TUSR -Nusername -Ddbname
```
To generate DDL for all users:

```
ddlgen -U login -P password -S server:port -TUSR -N dbname.%
```

**Example 37 Logical cluster** – generates DDL for “my_lcluster” on server “ase1”, enter:

```
ddlgen -Usa -P -Sase1 -TLC -Nmy_lcluster
```

**Example 38 Logical cluster** – generates DDL for all logical clusters on server “ase1”, enter:

```
ddlgen -Usa -P -Sase1 -TLC -N%
```

**Example 39 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.owner.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 -U login -P password -S host_name:port -TWS
-N dbname.owner.webservice_name
```

To generate DDL for all user-defined Web services owned by all users in database `dbname`:

```
ddlgen -U login -P password -S host_name:port -TWS -N dbname.%.%
```

**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:
At the command line, invoke `ddlgen` using the `ddlgen.bat` shell script file (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 DLL instead of dependent DLL.

The following section provides more details about:

- “Hiding passwords in `ddlgen`” on page 187
- “Using `ddlgen` for encrypted columns” on page 187

---

**Option** | **Parameter** | **Required** | **Default**
---|---|---|---
-U | `username` | Yes | None
-P | `password` | Yes | None
-S | `host_name:port_number` | Yes | None
-T | `object_type`<br>See the `-T` parameter description for a list of valid object types | No | Database
-N | `object_name` | Yes, if `object_type` for `-T` is not DB (database) | Default database name of `username`, if `-T object_type` is `db` or if `-T` is not specified
-D | `database_name` | No | Default database of `username`
-X | `extended_object_type`<br>Options are:<br>• OU – for user tables, user databases (excluding temporary databases), and stored procedures (excluding SQLJ procedures).<br>• OD – for proxy tables, temporary databases, and SQLJ procedures. | No; use only when the `object_type` for `-T` is:<br>• U (user table)<br>• P (procedure)<br>• DB (database) | None
-O | `output_file_name` | No | Standard out
-E | `error_file_name` | No | Standard out
-L | `log_file_name` | No | None
-V | `version_number` of `ddlgen` | No | None

---

- Option Parameter Required Default
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.

`ddlgen -P` password parameter option 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:

```bash
ddlgen -U login -P pwd -S server:port -T type -N name << 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:

```bash
ddlgen -U login -P pwd -S server:port -T type -N name << END
'cat filename'
END
```

Using `ddlgen` for encrypted columns

- The `ddlgen` utility supports pre-15.0.2 encryption. 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.

If you use `ddlgen` to generate DDL for encryption keys on Adaptive Server version 15.0.2 or later, the DDL may cause errors on a pre-15.0.2 version Adaptive Server, specifically if an encryption key is encrypted by a user specified-password or has key copies.

- The type `EK`, used for encryption key, generates the DDL to create an encryption key and to grant permissions on it. `ddlgen` generates encrypted column information and a `grant decrypt` statement, along with the table definition.

- If you do not specify the `-XOD` option, and the key to be migrated has been created in the source database using the `with passwd` clause, `ddlgen` generates a `create encryption key` command with `password` as its explicit password. This is similar to what `ddlgen` does for roles and login passwords.
The -XOD generates the create encryption key that specifies the key’s encrypted value as represented in sysencryptkeys. Use the -XOD to synchronize encryption keys across servers for data movement.

ddlgen -XOD generates DDL that includes a system encryption password (if it was set and DDL is generated for a key encrypted with a system encryption password) and DDL for keys.

Generating DDL without specifying the -XOD flag

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:

```
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
```

Generating DDL with the -XOD flag

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 -s -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 -S server -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 -S server -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 -S server -TU -N employee
```

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_charset]
    [-l interfaces_file]
    [-J [client_charset]]
    [-k keytab_file]
    [-P password]
    [-P 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.

-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, 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.

directory

copies the name of the operating system file destination or source for the
definition copy. The copy out overwrites any existing file.

database_name

copies 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

copies 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.
defncopy

- 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**  
**UNIX platforms only**  Allows you to view and edit server entries in the interfaces file from the command line in UNIX platforms. Located in $SYBASE/$SYBASE_OCS/bin.

**Syntax**  
dscp [-p]

or

dscp -v

To exit from dscp:

quit

or

exit

**Parameters**

-p  
suppresses command-line prompts.

-v  
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:

dscp -p

**Usage**  
- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use dscp.

- The dscp utility program is a text-based utility.

- See Chapter 5, “Using dscp” for more information about the dscp utility program.

**See also**  
Utilities  dsedit
dsedit

Description

**UNIX platforms**  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 `$SYBASE/$SYBASE_OCS/bin`.

**Windows**  The `dsedit.exe` utility creates and modifies network connection information in the interfaces file. The utility is located in `%SYBASE%\%SYBASE_OCS%\bin`.

Syntax

dsedit

or

dsedit -v

Parameters

- `-v` displays the version and copyright message of `dsedit`.

Usage

- You must set the SYBASE environment variable to the location of the current version of Adaptive Server before you can use `dsedit`.

- You must set the DISPLAY environment variable before invoking `dsedit`, unless you are only using the `-v` parameter to display the version number.

- For more information about the `dsedit` utility program, see Chapter 6, “Using dsedit” Also see the installation guide, and the configuration guide for your platform.

See also

Utilities  `dscp`
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

extractjava -j employees -f '/home/usera/jars/addr.jar' -new

• Windows:
extractjava -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
Installs a JAR from a client file into an Adaptive Server. The utility is located in $SYBASE/$SYBASE_OCS/bin.

Windows
The utility is instjava.exe, located in %SYBASE%%SYBASE_OCS%%bin.

Syntax
installjava

```
-f file_name
[-new | -update ]
[-j jar_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
installjava -v

Parameters

- `f file_name`
  is the name of the source file containing the classes to be installed in the database.

- `new | -update`
  specifies whether the classes in the file already exist in the database. If you specify:
  - `new` – you cannot install a class with the same name as an existing class
  - `update` – you can install a class with the same name as an existing class, and the newly installed class replaces the existing class

- `j jar_name`
  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.

- `S server_name`
  is the name of the server.

- `U user_name`
  is an Adaptive Server login name. If you omit the -U flag and parameter, or if you specify the -U flag with no parameter.
installjava

-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**

**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. -X enables both extended password encrypted connections and password encrypted connections without plain text password reconnection. 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.

-W disables both extended password and password encrypted negotiations

-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:

```sql
isql -A 4096
```
- To check your network packet size, enter:
  ```sql
  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.

-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, 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

-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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>r filename</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>R filename</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>use database_name</code></td>
<td>Changes the current database.</td>
</tr>
<tr>
<td><code>!! os_command</code></td>
<td>Executes an operating system command. Place at the start of a line.</td>
</tr>
</tbody>
</table>
| `file_name` | Redirects the output of the T-SQL command to `file_name`. The following example inserts the server version into `file_name`:
```
select @@version
go > file_name
```
| `>> file_name` | Appends the output of the T-SQL command to `file_name`. The following example appends the server version to `file_name`:
```
select @@version
go >> file_name
```
| `command` | 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 `sp_who`:
```
sp_who
go | grep sa
```
| `vi (UNIX)` | Calls the default editor. |
| `edit (Windows)` | |
| `reset` | Clears the query buffer. |
| `quit or exit` | 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.
- When you include the `-X` parameter, the password-enabled connection proceeds according to server capabilities:
  - If the server can handle both extended password and password encryption, extended password encryption negotiations are used.
  - If the server can handle password encryption only, password encryption negotiations are used.
• If the server cannot handle password encryption or extended password encryption, the first connection attempt fails and the client attempts to reconnect using a plain text password.

• 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:

```sql
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:

```sql
isql -c"." -csend
```

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:

```sql
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_bindexeclass, sp_configure, sp_defaultlanguage, sp_droplanguage, sp_helplanguage, sp_processmail, sp_remoteoption, sp_serveroption, sp_showcontrolinfo, sp_unbindexeclass, sp_volchanged
**langinstall**

**Description**
Installs a new language in an Adaptive Server. Located in $SYBASE/$SYBASE_ASE/bin.

**Windows**
The utility is langinst.exe, located in %SYBASE%\%SYBASE_ASE%\bin.

**Syntax**
langinstall

[-S server]
[-U user]
[-P password]
[-R release_number]
[-I path] language character_set

Or

langinstall -v

**Parameters**

- **-S server**
specifies the name of the Adaptive Server to which to connect. If you do not specify -S, langinstall uses the server specified by your DSQUERY environment variable. If DSQUERY is not set, langinstall attempts to connect to a server named SYBASE.

- **-U user**
specifies a login name. Login names are case sensitive.

- **-P password**
specifies the System Administrator’s (“sa” account) password. If you omit -P, langinstall prompts for the “sa” account password.

- **-R release_number**
specifies the release number, in the format \n.n.n, to use to upgrade messages in master..sysmessages. Use -R only in failure conditions, such as if langinstall (langinst in Windows) fails, in case of user error, or when you think that messages in sysmessages are out of date.

The -R parameter forces langinstall to collect messages from a release previous to the current one. langinstall compares the existing messages with the ones to be installed and replaces any that have changed.

For example, if the current version is 15.0, and the previous version was 12.5, and you think sysmessages may not be correct, include the messages from the earlier version in the syslanguages.upgrade column (12.5 in this case) by specifying -R 12.5. langinstall then installs all messages from Adaptive Server 12.5.
-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.
Tables used
master.dbo.syslanguages, master.dbo.sysmessages

See also
System procedures  sp_addlanguage, sp_addlogin, sp_configure,
sp_defaultlanguage, sp_droplanguage, sp_helplanguage

Utilities  defncopy, srvbuild
**optdiag**

**Description**
Displays optimizer statistics or loads updated statistics into system tables. `optdiag` is located in `$SYBASE/$SYBASE_ASE/bin`.

**Windows**
The utility is `optdiag.exe`, located in `%SYBASE%\%SYBASE_ASE%\bin`.

**Syntax**
```
optdiag [binary] [simulate] statistics
   { -i input_file [ database[.owner[.[table[.column]]]] ] [-o output_file] }
   [-U user_name]
   [-P password]
   [-T trace_value]
   [-l interfaces_file]
   [-S server]
   [-v]
   [-h]
   [-s]
   [-z language]
   [-J client_character_set]
   [-a display_charset]
```

**Parameters**
- **binary**
  extracts statistics in human-readable form and in binary form. When used with an input file (`-i input_file`), loads binary statistics into system tables.

- **simulate**
  specifies that `optdiag` display or load simulated statistics. See the `Performance and Tuning Guide`.

- **-i input_file**
  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”).

- **database**
  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.

- **owner**
  is the name of a table owner.
  - 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.
  - In input mode, `optdiag` ignores the table owner specified on the command line and uses the value in the input file.
**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:

<table>
<thead>
<tr>
<th>Flag value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Display status message “Next table is <code>table_name</code>” when in input mode.</td>
</tr>
<tr>
<td>4</td>
<td>Skip consistency checking for step numbers while loading histograms in input mode.</td>
</tr>
<tr>
<td>6</td>
<td>Display lines of input file during input mode. This flag has no effect in display mode.</td>
</tr>
<tr>
<td>7</td>
<td>Do not stop with a warning if the <code>optdiag</code> input file does not include sampling percent information.</td>
</tr>
</tbody>
</table>
-I interfaces_file
   specifies the name and location of the interfaces file to use when connecting
to Adaptive Server.

   If you do not use -I 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 (#s) 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**
Performs tests on an installation or database to determine its readiness for upgrade, and reports found problems. `preupgrade` is located in `$SYBASE/$SYBASE_ASE/upgrade`.

**Windows** The utility is `preupgrd.exe`, located in `%SYBASE%/%SYBASE_ASE%upgrade`.

**Syntax**
```
preupgrade [-v] [-h] [-N]
    [-p [skip_sybprocs]]
    [-D database_name]
    [-I interfaces_file]
    [-P password]
    [-S server_name]
    [-U user_name]
    [-X option[,option]...]
```

**Parameters**
- `-D database_name` limits checking to the named database and a subset of possible checks. Use this parameter to check newly loaded databases before bringing them online.
- `-h` prints help text and then exits.
- `-I interfaces_file` specifies an interfaces file for the server. The default is `$SYBASE/interfaces`.
- `-N` specifies `preupgrade` is to run in noninteractive mode. Thus, if `preupgrade` determines that any database is too small, the utility exits immediately.
- `-p [skip_sybprocs]` specifies whether you want to skip the parsing validity test on stored procedures.

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 `preupgrade -p` are:

- `-p` – skips the parsing in all databases
- `-p skip_procs` – skips parsing in `sybsystemprocs` while parsing the other databases
- Not using `-p` – parses text everywhere. This is the default.
-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>
<thead>
<tr>
<th>Check options</th>
<th>Options used with the -D parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td></td>
<td>Performs all permitted checks. When used with the -D option, only checks subset of options. Otherwise, all options are checked.</td>
</tr>
</tbody>
</table>

Table 7-3: Check options with the -X parameter
## Utility Commands Reference

<table>
<thead>
<tr>
<th>Check options</th>
<th>Options used with the <code>-D</code> parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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:  
  - 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.  
  - Warnings – occur when the current value of a parameter is between the maximum and minimum values, but less than the default value. |
| 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.  
  - Adaptive Server 15.0 and later reserves objects IDs 1 – 255.  
  - Adaptive Server 12.5.x and earlier reserves object IDs 1 – 99.  
  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 sybsystemdb. |
| 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.sysservers. 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. |
preupgrade

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 libtel.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:
0x01312a775ab9d5c71f99f05f7712d2cdec288d0ae1ce79268d0e8669313d1bc4c706

Replace the last part of the LDAP URL in libtel.cfg with this encrypted password:

ldap=libsybdldap.so
ldap://dolly:389/dc=sybase,dc=com????bindname=cn=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cdec288d0ae1ce79268d0e8669313d1bc4c706

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 libtel.cfg file.
qptune

Description
QPTune is an Adaptive Server utility written in Java/XML. It enables users to fix missing statistics and identify the best query plan, optimization goals, or other configuration settings, and apply them at the query or server level. This results in optimal performance of subsequent query executions.

Syntax
QPTune
[-U username]
[-P password]
[-S hostname:port/database]
[-A <action [start|collect(_full)|compare|fix|(start|collect|fix|undo_fix)_stats]>]
[-M mode]
[-T appTime]
[-i inputFile]
[-o outputFile]
[-f filelist]}
[-c configFile]
[-l limit]
[-e evalField]
[-d <diff%(|,diff_abs)>]
[-m missingCount]
[-n login]
[-J charset>]
[-N (noexec)]
[-g (applyOptgoal)]
[-v (verbose)]
[-s (sort)]
[-h (help)]

Parameters
-U username
  specifies the database user name.

-P password
  specifies the database password.

-S server
  specifies the database server. The database server is denoted by
  host:port/database.

Note
You must specify the -S option while using any QPTune action.
-A action
  specifies the action to be taken. Valid actions are:
  • start
  • collect
  • collect_full
  • compare
  • fix
  • start_stats
  • collect_stats
  • fix_stats
  • undo_fix_stats

-J charset
  specifies the character set used to connect to Adaptive Server. If this option
  is not specified, the Adaptive Server uses the server’s default character set.

  Note  If the installed JRE does not support the server's default charset
  encoding, you see an error message during the login process. Use the -J option
  to specify a more generic character set, such as -J utf8.

-M mode
  specifies the optimization goal or custom mode for an application. One of:
  allrows_oltp, allrows_dss, allrows_mix. You may also define custom modes;
  _basic_ is a system reserved custom mode.

-T appTime
  specifies the application running time, in minutes.

-o outputFile
  specifies the output file.

-i inputFile
  specifies the input file for the fix, fix_stats, and undo_fix_stats actions. You
  can also use -i to apply special rules to the specified queries for start for
  custom modes.

-f fileList
  compares a list of files to get the best plans; use commas to separate
  filenames.
-c configfile
  specifies the configuration file.

-l limit
  specifies a limit on the number of queries that should be analyzed and
  applied with special rules.

-e evalField
  is the evaluation field used for performance comparison.

-d difference
  specifies the percentage and absolute value difference for performance
  improvement to be considered outstanding.

-N used along with fix_stats and undo_fix_stats. -N generates a SQL script with
  update statistics or delete statistics statements. The update or delete
  statements are not executed through QPTune. The statements are written
  into a SQL script that is specified by the -o option.

-n login
  specifies the user's login whose query executions are collected and
  analyzed.

-m missingCount
  specifies the threshold value for missing statistics. The default value is 5.

-v
  specifies verbose mode.

-g
  when used along with the fix action, applies the default goal. The default goal
  is the best optgoal setting that most queries used as the best plan using
  QPTune's fix action. This option only generates plans for queries that do not
  currently use the server's default optimization goal.

Examples

Example 1 Fixes missing statistics, start the utility with the start_stats action:

QPTune -A start_stats -S my_host:4816/my_database
  -v

Executing : QPTune -U sa -P [unshown]
-S jdbc:sybase:Tds:my_host:4816/my_database
- A start_stats -M allrows_dss -T 0 -i null
-o metrics.xml -f null -c config.xml -l 5
-e elap_avg -d 5,5 -m 5 -n null -v
You are now connected to database: my_database
[INFO] Config: sp_configure 'capture missing
statistics', 1
Example 2  Uses collect_stats to retrieve missing statistics information from the sysstatistics table for statistics that exceed a specified threshold for count of missing statistics:

QPTune -A collect_stats -m 1 -o missingstats.xml -v
-S my_host:4816/my_database

Executing: QPTune -U sa -P [unshown] -S
jdbc:sybase:Tds:my_host:4816/my_database -A
collect_stats -M allrows_dss -T 0 -i null -o
missingstats.xml -f null -c config.xml -l 5 -e elap_avg
-d 5,5 -m 1 -n null -v
You are now connected to database: my_database
Now collecting missing statistics information from syssubtable on "Fri Sep 26 10:08:06 PDT 2008".
<?xml version="1.0" encoding="UTF-8"?>
<server url="jdbc:sybase:Tds:my_host:4816/my_database"
file="missingstats.xml"
type="missing stats" datetime="Fri Sep 26 10:08:06 PDT 2008">
</server>
The missing statistics information is written into XML file: missingstats.xml
[INFO] End config: sp_configure 'enable metrics
capture', 0
[INFO] End config: sp_configure 'abstract plan dump', 0
[INFO] End config: sp_configure 'system table', 0
[INFO] End config: sp_configure 'capture missing
statistics', 0
Program has restored the data source for metrics
collection.
----- QPTune finished executing. ------

Example 3  After collecting missing statistics information into an XML file
called missingstats.xml, updates the statistics using the fix_stats action:

QPTune -A fix_stats -m 1 -i missingstats.xml
         -v -S my_host:4816/my_database
Executing : QPTune -U sa -P [unshown] -S
jdbc:sybase:Tds:my_host:4816/my_database -A fix_stats -
M allrows_dss -T 0 -i missingstats.xml -o metrics.xml -
f null -c config.xml -l 5 -e elap_avg -d 5,5 -m 1 -n
null -v
You are now connected to database: my_database
Fix statistics on "Fri Sep 26 10:14:59 PDT 2008"
------------------------------------------------------
-----
Details of statements(s) fixed:
-------------------------------
Fixed statistics:[Update] Y(y4,y2)
[INFO] Fix Statement = update statistics Y(y4,y2)
Fixed statistics:[Update] Y(y3)
[INFO] Fix Statement = update statistics Y(y3)
Fixed statistics:[Update] Y(y2,y1)
[INFO] Fix Statement = update statistics Y(y2,y1)
Fixed statistics:[Update] Y(y1)
[INFO] Fix Statement = update statistics Y(y1)
----- QPTune finished executing. ------

Generates a SQL script for updating statistics, without executing the actual
updates, by using the -N option to indicate “noexec”, and the -o option to
indicate the output script file:

QPTune -U sa -P -S my_host:5000/my_database
         -A fix_stats -m 5 -i missingstats.xml
         -N -o missingstats.sql

Example 4  Starts QPTune to apply standard optimization goal settings to
queries:

QPTune -S host:port/database -A start
         [-M {allrows_oltp, allrows_dss, allrows_mix}]
Start QPTune to apply custom rules to specified queries:

```
QPTune -S host:port/database -A start -M custom_1
       -i input.xml -l 3 [-v]
```

**Example 5** Runs your application and collect metrics into an XML file named `a2.xml`:

```
QPTune -S host:port/database -A collect -T 0
       -o a2.xml -v
```

Program has configured the data source for metrics collection. Now collecting information from sysquerymetrics on "Tue Feb 19 22:16:04 PST 2008".

```xml
<?xml version="1.0" encoding="UTF-8"?>
<server url="jdbc:sybase:Tds:SHANGHI:5000" type="ASE" mode="custom_1"
datetime="Tue Feb 19 22:16:04 PST 2008">
    <query id="1">
        <qtext>select count(T.title_id) from authors A, titleauthor T
             where A.au_id = T.au_id </qtext>
        <elap_avg>300</elap_avg>
        <bestmode>custom_1</bestmode>
    </query>
</server>
```

**Example 6** Once metrics are collected, compares different XML files to get the best query optimization goal or criteria for each of the queries:

```
QPTune -A compare -f a1.xml,a2.xml -d 51,10
       -o best.xml -S my_host:5000/my_database
```

This result shows a comparison between two XML metrics files: `a1.xml` has six queries, and `a2.xml` has seven queries. Comparisons can only be made between the queries that are common to both files. There are three queries that ran faster in `a2.xml`:

```
Compare all the files: | a1.xml, a2.xml|
Report generated on "Tue Aug 19 21:13:04 PST 2008"
--------------------------------------------------------------------------
File #1: [name= a1.xml : mode=allrows_mix]
File #2: [name= a2.xml : mode=custom_1]
Query count in File #1 : [mode=allrows_mix] 6
Query count in File #2 : [mode=custom_1] 7
==========================================================================
Query count improved in File #2: [mode=allrows_mix] 3
```

Total performance improved [from 422 to 129]: 69 %
Following queries run better in File #2:

\[\text{mode=allrows_mix}\]

------------------------------------------------------------------------

Group 1: improved by no more than 25% [0 queries]
Group 2: improved by 25% to 50% [1 queries]
Query: \(\text{select count(T.title_id) from authors A, titleauthors T where A.au_id = T.au_id}\)
Average elapsed time (ms): File #1=100  File #2=50   Improvement=50.0%
Outstanding=No
Group 3: improved by 50% to 75% [0 queries]
Group 4: improved by 75% to 100% [2 queries]
Query: \(\text{select count(*) from titles T, titleauthors TA where T.title_id = TA.title_id}\)
Average elapsed time (ms): File #1=34  File #2=7       Improvement=79.0%
Outstanding=Yes
Query: \(\text{select au_lname, au_fname from authors where state in ("CA", "AZ")}\)
Average elapsed time (ms): File #1=9   File #2=0       Improvement=100.0%
Outstanding=No

Usage

If specific values are not indicated for the parameters, the following defaults are used:

- \(\text{-A : collect}\)
- \(\text{-M : allrows_dss}\)
- \(\text{-T : 0}\)
- \(\text{-o : metrics.xml}\)
- \(\text{-c : config.xml}\)
- \(\text{-e : elap_avg}\)
- \(\text{-d : 5.5. If percentage is specified but not the absolute value, then absolute value defaults to 0.}\)
- \(\text{-l limit}\)
- \(\text{-m 5}\)

Permissions

QPtune’s compare action may be run by any user. All other actions of QPTune may only be run by users with sa_role and sso_role.

See also

For more information about the QPTune utility or the QPTune GUI, see the Migration Technology Guide.
qrmutil

Description

Cluster Edition only – qrmutil is a command line utility that allows you to back up, restore, and reconfigure the quorum device. qrmutil is located in SYBASE/SYBASE_ASE/bin.

Syntax

```bash
--additional-run-parameters=parameter_list
--ase-config-extract=file_name
--ase-config-info
--ase-config-store=file_name
--ase-config-version=version_number
--buildquorum=[force]--cluster-take-over
--config-file=file_name
--diag={all | boot | toc | nodes | locks | config | cms}
--display={boot | nodes | heartbeat | master | cluster | instance | config | state}
--drop-cluster=[force]
--drop-instance=instance_name
--errorlog=file_name
--extract-config=file_name
-h, --help
-F, --cluster-input=file_name
--fence-capable=device_path
--installation=installation_mode
-s, --instance=instance_name
--instance-node=node_name
--interfaces-dir=path_to_interfaces_file
--max-instances=number_of_instances
--master-dev=master_device
--membership-mode=membership_mode
--primary-address=interconnect_address
--primary-port=port_number
--primary-protocol=protocol
-Q, --quorum-dev=quorum_device
--register-node=node_name
--secondary-address=interconnect_address
--secondary-port=port_number
--secondary-protocol=protocol
--traceflags=traceflag_list
--unregister-node=node_name
--verify-node=node_name
-v, --version]
```

Parameters

--additional-run-parameters=parameter_list

parameters that Unified Agent uses to start the data server. Unlike other settings, dataserver does not read the additional run parameters. They are read by the unified agent and passed to the dataserver command line. If you include the --instance parameter, the additional run parameters apply to the specified instance. Otherwise, the additional run parameters apply to all instances in the cluster.
--ase-config-extract=\textit{file\_name}
extracts the Adaptive Server configuration file stored on the quorum device to the named file.

--ase-config-info
displays information about the Adaptive Server configuration file stored on the quorum device.

--ase-config-store=\textit{file\_name}
stores the named file in the quorum device as the Adaptive Server configuration file.

--ase-config-version=[\textit{version\_number}]
displays or sets the version of the master Adaptive Server configuration file stored on the quorum device.

--buildquorum[=force]
builds a new quorum device. Use =force to overwrite an exiting file or an existing quorum device on a raw partition. You must include the --cluster-input parameter with --buildquorum.

--config-file=\textit{config\_file\_name}
if used with -instance, sets this path to the Adaptive Server configuration file for the specified instance. If you do not include -instance, sets the path to the cluster-wide configuration file.

--diag={all | boot | toc | nodes | locks | config | cms}
for internal use only.
--display={boot | nodes | heartbeat | master | cluster | instance | config | state}
displays the current state of cluster or instance:

- boot – displays start-up information for the cluster, including the
  version of the quorum device, any trace flags issued at start-up, the boot
  ID of the cluster, and any messages displayed at start-up.
- nodes – displays the registered management nodes.
- heartbeat – displays heartbeat information for all nodes in the cluster.
- master – displays master device information.
- cluster – displays the cluster configuration.
- instance – displays the instance configuration. You must include
  --instance=instance_name with this parameter.
- config – displays configuration for the cluster and for all instances in the
  cluster.
- state – displays the current state for the cluster and for all instances in
  the cluster.

--drop-cluster=[force]
drops a cluster and removes the quorum device. Use =force to force the drop
if the quorum indicates the cluster is running.

Warning! --drop-cluster removes the cluster.

--drop-instance=instance_name
Sybase internal use only.

Warning! Use the sybcluster utility to drop an instance from the cluster.

--errorlog=log_file_name
full path to the error log for the specified instance. You must include the
instance-name parameter. Takes effect at next restart of the instance.

--extract-config=file_name
extracts the configuration area of the quorum device to the specified file.

-h | --help
displays the full syntax of qrmutil.

-F | cluster-input=file_name
loads the cluster configuration from the specified cluster input file.
--fence-capable=device_path
    tests if specified device can be used for I/O fencing. Returns either “Device
    is fence capable” or “Device is not fence capable”.

--installation=installation_mode
    changes the installation mode for the cluster. Values are:
    • shared (default)
    • private

-instance=instance_name
    applies qrmutil parameters to a specified instance.

-interfaces-dir=interfaces_path
    the path to a directory that contains a file named interfaces. If this parameter
    is used with --instance, it sets the path to the interfaces file for the specified
    instance. If --instance is not included, sets the path to the cluster-wide
    interfaces file.

--max-instances=number_of_instances
    sets the maximum number of instances for the cluster configuration.

--master-dev=master_device_name
    changes the master device the cluster uses.

--membership-mode=membership_mode
    sets the membership mode. Values are:
    • native (default)
    • vcs

--primary-address=interconnect_address
    changes the primary interconnect address for a given instance.

--primary-port=port_number
    changes the starting port number for the primary interconnect for a given
    instance.

--primary-protocol=protocol
    changes the protocol used for the primary cluster interconnect.

-Q | --quorum-dev=quorum_path
    specifies the full path to the quorum device.

--register-node=node_name
    registers a node for quorum management.
--secondary-address=inteconnect_address
changes the secondary interconnect address for a given instance.

--secondary-port=port_number
changes the starting port number for the secondary interconnect for a given instance.

--secondary-protocol=protocol
changes the protocol used for the secondary cluster interconnect.

--traceflags=trace_flag, trace_flag
changes the cluster-wide or the instance-specific trace flags for start-up. If you do not include a list of trace flags, qrmutil clears the trace flags for the cluster instance.

--unregister-node=node_name
unregisters a node from quorum management.

--verify-node=node_name
indicates that the specified node is registered on the quorum device.

-v | --version
displays the version information for the qrmutil utility.

Examples

**Example 1** This example changes the path to the error log to /sybase/opt/cluster/ASE-15_0/ase1.log:

```
qrmutil --quorum_dev=/dev/raw/raw101 --instance=ase1 --errorlog=/sybase/cluster/ASE-15_0/ASE-15_0/ase1.log
```

**Example 2** This example registers the node “blade5” for mycluster:

```
qrmutil --quorum_dev=/dev/raw/raw101 --register-node=blade5
```

**Example 3** This example creates a new quorum device for the cluster “mycluster”:

```
qrmutil --quorum-dev=/dev/raw/raw101 --cluster-input=/sybase/cluster/ase1.inp -buildquorum
```

**Example 4** This example backs up the quorum device to /sybase/cluster_bak/quorum.bak:

```
qrmutil --quorum-dev=/dev/raw/raw101 --extract-config=/sybase/cluster_bak/quorum.bak
```

**Example 5** This example restores the quorum device from the backup created in /sybase/cluster_bak/quorum.bak:

```
qrmutil --quorum-dev=/dev/raw/raw101 --cluster-input=/sybase/cluster_bak/quorum.bak --buildquorum=force
```
Example 6  This example displays the cluster configuration stored on the quorum device:

```
qrmutil --quorum-dev=/dev/raw/raw101 --display=config
```

Example 7  This example tests whether the named device can be fenced:

```
qrmutil --quorum-dev=/dev/raw/raw101 --fence-capable=/dev/raw/raw106
```

Usage

- qrmutil is primarily a diagnostic utility. Sybase recommends that you use sybcluster to make configuration changes to the cluster.
- You can pass as many as 20 commands to qrmutil. However, you can specify the --instance= parameter only once.
- If you specify --buildquorum, the quorum is built and qrmutil exits without running any commands other than --cluster-input.
- qrmutil exits after it executes the --drop-cluster parameter.
- This is an example of using multiple commands:

```
qrmutil --quorum-dev=/dev/raw/raw101 --display=cluster
   --register-node=blade1 --unregister-node=blade2 --verify-node=blade3
```

Permissions

To run qrmutil, you must be the same sybase user that started the instance, with execute privileges on the qrmutil binary, have direct access to the quorum device, and at least read permissions on the quorum 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
```

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>%CPU</th>
<th>%MEM</th>
<th>SZ</th>
<th>RSS</th>
<th>TT</th>
<th>STAT</th>
<th>START</th>
<th>TIME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>user114276</td>
<td>0.0</td>
<td>1.7</td>
<td>712</td>
<td>1000</td>
<td>?</td>
<td>S</td>
<td>Apr</td>
<td>5514:05</td>
<td>dataserver -d greensrv.dat -sgreensrv -e/install/greensrv+_errorlog</td>
<td></td>
</tr>
<tr>
<td>sybase</td>
<td>1071</td>
<td>0.0</td>
<td>1.4</td>
<td>408</td>
<td>820</td>
<td>?</td>
<td>S</td>
<td>Mar</td>
<td>28895:38</td>
<td>/usr/local/sybase/bin/dataserver -d/dev/rsd1f -e/install/errorlog</td>
</tr>
<tr>
<td>user128493</td>
<td>0.0</td>
<td>0.0</td>
<td>3692</td>
<td></td>
<td></td>
<td>IW</td>
<td>Apr</td>
<td>1</td>
<td>0:10</td>
<td>backupserver -SSYB_BACKUP -e/install/backup.log -I/interfaces -M/bin/sybmultbuf -Lus_english -Jiso_1</td>
</tr>
</tbody>
</table>

**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`
sqldbgr

Description

sqldbgr is a command-line utility that debugs stored procedures and triggers. As with many source-level debuggers, you can:

- attach sqldbgr to a task
- set, enable, and disable breakpoints
- step through a task one line at a time
- step into and out of procedures
- detach sqldbgr from stored procedures or triggers once the debugging is complete.

UNIX platforms  sqldbgr is located in $SYBASE/$SYBASE_ASE/bin.

Windows  sqldbgr is located in %SYBASE%\%SYBASE_ASE%\bin.

Note  You do not have the ability to view sqldbgr version strings.

Syntax

sqldbgr

-U username
-P password
-S host:port

Parameters

-U username
speifies the user name. You must insert a space between -U and username.

-P password
speifies the user password. You must insert a space between -P and password.

-S host:port
speifies the machine name and the port number. You must insert a space between -S and host:port.

Examples

Example 1  Shows sqldbgr debugging stored procedures and triggers on host MERCURY:

$SYBASE/$SYBASE_ASE/bin/sqldbgr -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
sqldbg

<table>
<thead>
<tr>
<th>fid</th>
<th>spid</th>
<th>status</th>
<th>loginame</th>
<th>origname</th>
<th>hostname</th>
<th>blk_spid</th>
<th>dbname</th>
<th>cmd</th>
<th>block_xloid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>NETWORK HANDLER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>NETWORK HANDLER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>DEADLOCK TUNE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>MIRROR HANDLER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>ASTC HANDLER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>ASTC HANDLER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>CHECKPOINT SLEEP</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>0</td>
<td>HOUSEKEEPER</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>running</td>
<td>sa</td>
<td>sa</td>
<td>master</td>
<td>0</td>
<td>SELECT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>sleeping</td>
<td>sa</td>
<td>sa</td>
<td>master</td>
<td>0</td>
<td>SELECT</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(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 sqldbg 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
go
------
12
1>
$SYBASE/$SYBASE_ASE/bin/sqldbg -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 sqldbgr 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 sqldbgr, you must set either the SYBASE_JRE or JAVA_HOME environments to the location containing the Java run environment.

- When you invoke sqldbgr at the command prompt, the utility starts and the prompt changes to a sqldbgr prompt:

  (sqldbgr)

Once you see the (sqldbgr) prompt, you can enter the following sqldbgr commands to perform your tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attach spid</td>
<td>Attaches a task to sqldbgr when you are already logged in to Adaptive Server.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Do not use attach spid to attach to a procedure that is not running.</td>
</tr>
<tr>
<td></td>
<td>sqldbgr cannot debug multiple tasks in the same session. If you try to attach</td>
</tr>
<tr>
<td></td>
<td>the utility to multiple tasks, the first spid continues to be marked as</td>
</tr>
<tr>
<td></td>
<td>attached. Since you cannot attach to a spid that is already attached, you</td>
</tr>
<tr>
<td></td>
<td>must use the detach command, and then attach to another spid.</td>
</tr>
<tr>
<td>run procname</td>
<td>Debugs stored procedures and triggers without attaching sqldbgr to an existing task.</td>
</tr>
<tr>
<td></td>
<td>If you attempt to use run proname while you are already debugging an existing task with attach spid, run proname fails and you see the following:</td>
</tr>
<tr>
<td></td>
<td>Cannot run a procedure while debugging another task</td>
</tr>
<tr>
<td>stop in proname</td>
<td>Sets a breakpoint to stop the stored procedure or trigger being debugged at the beginning of the specified procedure name.</td>
</tr>
<tr>
<td>[at line #]</td>
<td>stop in proname [at line #] sets a breakpoint to stop the stored procedure or trigger being debugged at a designated line within the specified procedure.</td>
</tr>
<tr>
<td></td>
<td>If you enter an invalid line number, sqldbgr moves the breakpoint to the next valid line number, and displays:</td>
</tr>
<tr>
<td></td>
<td>Invalid line number</td>
</tr>
<tr>
<td></td>
<td>You can also use this command to set multiple breakpoints.</td>
</tr>
</tbody>
</table>
**Command** | **Description**  
--- | ---  
show breakpoints | 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. 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. An asterisk (*) in the breakpoint line indicates that the breakpoint is set, but currently disabled.  
use dbname | Tells sqldbgr what database to use in order to debug that database’s stored procedures or triggers.  
show variables [at level #] | show variables displays all the variables and their values in the current SQL stored procedure or trigger. 
show variables at level # | displays the variables and their values in the current SQL stored procedure or trigger at the specified level.  
show @varname [at level #] | show @varname displays the indicated variable and its value in the current SQL stored procedure or trigger. 
show @varname at level # | displays the indicated variable and its value in the current SQL stored procedure or trigger at the specified level.  
**Note**: sqldbgr does not support Java variables.  
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 @varname = 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 @varname = 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 any_sql_statement | Executes ad hoc SQL statements. You can use this command to select and analyze data from temp tables created by the task being debugged. sql any_sql_statement returns a result set and any errors that occurred.
### Table 7-5: sqldbgr error messages and their meaning

<table>
<thead>
<tr>
<th>Error message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot allocate resource in ASE</td>
<td>Indicates that Adaptive Server does not have sufficient memory resources to execute sqldbgr. Increase procedure cache size and restart sqldbgr.</td>
</tr>
<tr>
<td>Cannot create Debugger handle in ASE</td>
<td>Indicates that Adaptive Server does not have sufficient memory resources to create a debugger handle. Increase procedure cache size and restart sqldbgr.</td>
</tr>
<tr>
<td>The spid is invalid</td>
<td>Displays when you attempt to attach sqldbgr to an invalid spid. Double check the spid and try again.</td>
</tr>
<tr>
<td>You cannot debug a task that is not owned by you</td>
<td>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.</td>
</tr>
<tr>
<td>Spid is already being debugged</td>
<td>Displays when you execute attach spid and attempt to attach to a spid that is already being debugged.</td>
</tr>
<tr>
<td>Spid is not debugged currently</td>
<td>Displays when you execute detach spid and attempt to detach from a spid that is not attached to sqldbgr.</td>
</tr>
<tr>
<td>Invalid command</td>
<td>Displays when you enter an invalid command.</td>
</tr>
<tr>
<td>Invalid procedure name</td>
<td>Displays when you enter an invalid procedure name in stop in procname.</td>
</tr>
<tr>
<td>Invalid line number</td>
<td>Displays when you enter an invalid line number in stop in procname at line #.</td>
</tr>
<tr>
<td>Variable not found</td>
<td>Displays when you enter an invalid variable in show @varname, show @varname at level #, or set @varname = VALUE.</td>
</tr>
<tr>
<td>Illegal conversion attempted</td>
<td>Displays when you execute set @varname = VALUE and attempt to convert the variable to an invalid value.</td>
</tr>
<tr>
<td>Conversion from text to datatype failed</td>
<td>Displays when set @varname = VALUE is unsuccessful.</td>
</tr>
<tr>
<td>Cannot run a procedure while debugging another task</td>
<td>Displays if you use run proname while already debugging an existing task with attach spid.</td>
</tr>
</tbody>
</table>
sqlloc

Description

UNIX platforms only  Installs and modifies languages, character sets, and sort order defaults for Adaptive Server using a GUI based on X11/Motif. sqlloc is located in $SYBASE/$SYBASE_ASE/bin.

Syntax

sqlloc
    [-S server]
    [-U user]
    [-P password]
    [-s sybase dir]
    [-l interfaces file]
    [-r resource file]

Or

    sqlloc -v

Parameters

-S server
    specifies the name of the Adaptive Server to which to connect.

-U user
    specifies a login name. Logins are case sensitive.

-P password
    specifies the “sa” account password.

-s sybase dir
    specifies the value to use for the SYBASE environment variable.

-l interfaces file
    specifies the name and location of the interfaces file to search when connecting to Adaptive Server.

-r resource file
    executes the specified resource file.

-v
    prints the version number and copyright message for sqlloc 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 sqlloc.

- You must set the DISPLAY environment variable before invoking sqlloc, unless you are only using the -v parameter to display the version number.

Permissions

You must be a Sybase System Administrator to use sqlloc.

See also

Documentation  Installation Guide for UNIX Platforms for more information about sqlloc.

Utilities  langinstall, sqllocres
**sqllocres**

**Description**  
**UNIX platforms only** Installs and modifies languages, character sets, and sort order defaults for Adaptive Server, using a resource file. sqllocres is located in $SYBASE/$SYBASE_OCS/bin.

**Syntax**

```
sqllocres
    [-S server]
    [-U user]
    [-P password]
    [-s sybase dir]
    [-l interfaces file]
    [-r resource file]
```

Or

```
sqllocres -v
```

**Parameters**

- `-S server` specifies the name of the Adaptive Server to which to connect.
- `-U user` specifies a login name.
- `-P password` specifies the “sa” account password.
- `-s sybase dir` specifies the value to use for the SYBASE environment variable.
- `-l interfaces file` specifies the name and location of the interfaces file to search when connecting to Adaptive Server.
- `-r resource file` 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 Installation Guide for UNIX Platforms.

**Utilities**  
langinstall, sqloc
sqlsrvr

Description
The executable form of the Adaptive Server program, this utility is located in %SYBASE\%SYBASE_ASE%\bin.

Syntax

Parameters

-f
forces initialization of a device or database. You must use both -b and -w to use -f.

-g
turns off event-logging.

-G
specifies the name of the event log server.

-h
prints this help message, then exits.

-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 sqlsrvr and 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.

-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 \texttt{sqlsrvr -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 \textit{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 \texttt{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, sqlsrvr looks for a file named
  \texttt{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:
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:

  `-C configuration_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

Commands  disk mirror, disk remirror, reconfigure
System procedures  sp_configure, sp_password
Utilities  startserver
sqlupgrade

**Description**

**UNIX platforms only**  Upgrades your currently installed version of Adaptive Server to the newest release using a GUI based on X11/Motif. `sqlupgrade` is located in `$SYBASE/$SYBASE_ASE/bin`.

**Syntax**

```
sqlupgrade
  [-s sybase dir]
  [-r resource file]

Or

sqlupgrade -v
```

**Parameters**

- `-s sybase dir`
  specifies the value to use for the SYBASE environment variable.

- `-r resource file`
  executes the specified resource file.

- `-v`
  prints the version number and copyright message for `sqlupgrade` 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 `sqlupgrade`.

- You must set the DISPLAY environment variable before invoking `sqlupgrade`, unless you are only using the `-v` parameter to display the version number.

**Permissions**

You must be a Sybase System Administrator to use `sqlupgrade`.

**See also**

**Documentation**  *Installation Guide for UNIX Platforms* for more information about `sqlupgrade`.

**Utilities**  `preupgrade`, `sqlupgraderes`
sqlupgraderes

Description
**UNIX platforms only** Upgrades your currently installed release of Adaptive Server to the newest release using resource files. sqlupgraderes is located in $SYBASE/$SYBASE_OCS/bin.

Syntax
```
sqlupgraderes
   [-s sybase dir]
   [-r resource file]

Or
   sqlupgraderes -v
```

Parameters
- `-s Sybase Dir`
  specifies the value to use for the SYBASE environment variable.

- `-r Resource File`
  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
**Documentation** *Installation Guide for UNIX Platforms* for more information about sqlupgraderes.

Utilities
  sqlupgrade
**srvbuild**

**Description**  
**UNIX platforms only**  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 srvbuild in either GUI mode based on X11/Motif, or non-GUI mode. srvbuild is located in $SYBASE/$SYBASE_ASE/bin.

**Syntax**  
srvbuild  
[-s sybase_dir]  
[-I interfaces_file]  
[-r resource_file]  
Or  
srvbuild -v

**Parameters**
- `-s sybase_dir`
  specifies the value to use for the SYBASE environment variable.
- `-I interfaces_file`
  specifies the name and location of the interfaces file to search when connecting to Adaptive Server.
- `-r resource_file`
  executes the specified resource file.
- `-v`
  prints the version number and copyright message for srvbuild 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 srvbuild.
- Before invoking srvbuild, unless you are only using the `-v` parameter to display the version number.

**Using LDAP with srvbuild in a 64-bit environment**

When you use srvbuild 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.

srvbuild is a 32-bit application and uses the LDAP server entry from the $SYBASE/$SYBASE_OCS/config/libtcl.cfg file. Adaptive Server is a 64-bit application and uses the LDAP server information from the $SYBASE/$SYBASE_OCS/config/libtcl64.cfg file.
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

**Utilities**  srvbuildres
**srvtbuildres**

**Description**

UNIX platforms only  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. *srvtbuildres* is located in $SYBASE/$SYBASE_ASE/bin.

**Syntax**

```
srvtbuildres
    [-s *sybase_dir*
        [-l *interfaces_file*]
        [-r *resource_file*]
    
    Or
    srvtbuildres -v
```

**Parameters**

- `-s *sybase_dir*`
  specifies the value to use for the SYBASE environment variable.

- `-l *interfaces_file*`
  specifies the name and location of the interfaces file to search when connecting to Adaptive Server.

- `-r *resource_file*`
  executes the specified resource file.

- `-v`
  prints the version number and copyright message for *srvtbuildres* 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 *srvtbuildres*.

**Permissions**

You must be a Sybase System Administrator to use *srvtbuildres*.

**See also**

Documentation  *Installation Guide for UNIX Platforms* for more information about *srvtbuildres*.

Utilities  *srvtbuild*
startserver

Description UNIX platforms only  Starts an Adaptive Server or a Backup Server.

startserver is located in $SYBASE/$SYBASE_ASE/bin.

Syntax

startserver [ [-f runserverfile] [-m]] ...

Parameters

- \( \text{-f runserverfile} \)
  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
  Run_servername. If you start a second Adaptive Server on the same
  machine, startserver creates a new runserver file named Run_servername.

- \( \text{-m} \)
  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 dbo use only parameter of
  sp_dboption 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.
  The -m parameter creates an m_RUNSERVER file and overwrites any
  existing m_RUNSERVER file.

Examples

Example 1 Starts an Adaptive Server named SYBASE from the runserver file
named Run_servername in the current directory:

  startserver

Example 2 Starts an Adaptive Server named MYSERVER and a Backup
Server named SYB_BACKUP:

  startserver -f RUN_MYSERVER -f RUN_SYB_BACKUP

Example 3 Starts only the Backup Server SYB_BACKUP:

  startserver -f RUN_SYB_BACKUP

Usage

- 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.
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/
#
/$SYBASE/$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:
```
startserver

# 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**

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:

**UNIX platforms**

$SYBASE/$SYBASE_ASE/bin/sybmigrate

**Windows**

%SYBASE%\%SYBASE_ASE%\bin\sybmigrate.bat

**Syntax**

sybmigrate [-v] [-h] [-f]

[-D 1 | 2 | 3 | 4]

[-l interfaces_file]

[-r input_resource_file]

[-m setup | migrate | validate | report]

[-m 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]

**Parameters**

- **-v**
  
  prints the version string and exits.

- **-h**
  
  prints the help information and syntax usage and exits.

- **-f**
  
  overrides the locking session.

- **-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, sybmigrate uses:

  - For UNIX – $SYBASE/interfaces
  
  - For Windows – %SYBASE%\ini\sql.ini

- **-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.
-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.

-rm
indicates what type of report to generate. If -rm 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 9, “Migration Utility” for detailed information on sybmigrate.
### sybtsmpasswd

**Description**  
Records or changes the user password and creates the Tivoli Storage Manager (TSM) encrypted password file, *TSM.PWD*, on the TSM client machine. The location of the file is the directory specified by the PASSWORDDIR configuration parameter in the TSM configuration file.

**Note**  
sybtsmpasswd is supported when the IBM Tivoli Storage Manager is licensed at your site.

**Syntax**  
sybtsmpasswd

**Examples**  
sybtsmpasswd prompts for password information.

```bash
sybtsmpasswd

Enter your current password:
Enter your new password:
Enter your new password again:

Your new password has been accepted and updated.
```

**Usage**  
- Execute sybtsmpasswd when you record or change the user password. Make sure the same user name and password are present in the *TSM.PWD* file on the TSM client node.

**Permissions**  
Only the operating system “root” user can execute sybtsmpasswd.
**xpserver**

**Description**
Starts XP Server manually.

**Syntax**
```
xpserver -S XP_Server
```

- `-S XP_Server`
  - Specifies the name of the XP Server to start. The format of the XP server name is `SQLSERVERNAME_XP`, where `SQLSERVERNAME` 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.

- `-I interfaces_file`
  - Specifies the name and location of the directory containing the interfaces file (`sql.ini`) that Adaptive Server searches when connecting to XP Server. If you do not specify `-I`, `xpserver` uses the `ini` subdirectory of the `%SYBASE%` release directory.

- `-p priority`
  - 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.

- `-s stack_size`
  - 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.

- `-u`
  - 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.

- `-v`
  - Prints the version number and copyright message for XP Server and then exits.
xpserver

-x

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.

Usage

- XP Server is normally started automatically by Adaptive Server. Use the manual command to start XP Server only when instructed to do so in an “XP Server Failed to Start” error message.
- 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.
- 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 in 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_bindexeclass.

See the discussion of multithread programming in the Open Server Server Library/C Reference Manual for information about scheduling Open Server threads.

- 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.
- Unlike Adaptive Server and Backup Server, XP Server does not have a runserver file.
- When configuring an XP Server, the directory service entry name must end with “_XP” in upper case, such as “abcdef_XP” or “ABCDEF_XP.”

Permissions

No special permissions are required to run xpserver.

See also

System ESP xp_cmdshell
System procedures sp_configure, sp_freedll
CHAPTER 8

Using sybcluster to Manage the Cluster

This chapter describes sybcluster, which is the command line utility that you can use to create, start, and manage a shared-disk cluster, and manage instances in the cluster. It describes the parameters and options available in sybcluster.

sybcluster is only available in a shared-disk cluster environment. For information about how to use sybcluster, see the Clusters Users Guide.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>sybcluster</td>
<td>280</td>
</tr>
<tr>
<td>sybcluster interactive commands</td>
<td>284</td>
</tr>
<tr>
<td>add backupserver</td>
<td>287</td>
</tr>
<tr>
<td>add instance</td>
<td>287</td>
</tr>
<tr>
<td>connect</td>
<td>290</td>
</tr>
<tr>
<td>create backupserver</td>
<td>291</td>
</tr>
<tr>
<td>create cluster</td>
<td>292</td>
</tr>
<tr>
<td>create monitorserver</td>
<td>294</td>
</tr>
<tr>
<td>create xpserver</td>
<td>295</td>
</tr>
<tr>
<td>deploy plugin</td>
<td>295</td>
</tr>
<tr>
<td>diagnose cluster</td>
<td>297</td>
</tr>
<tr>
<td>diagnose instance</td>
<td>298</td>
</tr>
<tr>
<td>disconnect</td>
<td>299</td>
</tr>
<tr>
<td>drop backupserver</td>
<td>299</td>
</tr>
<tr>
<td>drop cluster</td>
<td>300</td>
</tr>
<tr>
<td>drop instance</td>
<td>301</td>
</tr>
<tr>
<td>drop monitorserver</td>
<td>302</td>
</tr>
<tr>
<td>drop xpserver</td>
<td>302</td>
</tr>
<tr>
<td>exit</td>
<td>303</td>
</tr>
<tr>
<td>help</td>
<td>303</td>
</tr>
<tr>
<td>localize</td>
<td>303</td>
</tr>
<tr>
<td>quit</td>
<td>305</td>
</tr>
<tr>
<td>set backupserver</td>
<td>305</td>
</tr>
</tbody>
</table>
## sybcluster

### Description
Manages a Sybase shared-disk cluster. sybcluster lets you create, start, stop, and manage a cluster or any instance in a cluster.

### Syntax
```
sybcluster

[C cluster_name]
[d discovery_list]
[f agent_connection]
[h]
[i instance_name]
[l input_file_path]
[L]
[m message_level]
[P [password]]
[U user_name] (the default value is “uafadmin”)
[v]
```

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>set cluster</td>
<td>306</td>
</tr>
<tr>
<td>set instance</td>
<td>307</td>
</tr>
<tr>
<td>set monitorserver</td>
<td>308</td>
</tr>
<tr>
<td>set xpserver port</td>
<td>309</td>
</tr>
<tr>
<td>show agents</td>
<td>309</td>
</tr>
<tr>
<td>show backupserver config</td>
<td>311</td>
</tr>
<tr>
<td>show cluster</td>
<td>312</td>
</tr>
<tr>
<td>show instance</td>
<td>315</td>
</tr>
<tr>
<td>show membership mode</td>
<td>317</td>
</tr>
<tr>
<td>show monitorserver config</td>
<td>317</td>
</tr>
<tr>
<td>show session</td>
<td>318</td>
</tr>
<tr>
<td>show xpserver</td>
<td>320</td>
</tr>
<tr>
<td>shutdown cluster</td>
<td>320</td>
</tr>
<tr>
<td>shutdown instance</td>
<td>321</td>
</tr>
<tr>
<td>start cluster</td>
<td>322</td>
</tr>
<tr>
<td>start instance</td>
<td>322</td>
</tr>
<tr>
<td>upgrade server</td>
<td>323</td>
</tr>
<tr>
<td>use</td>
<td>325</td>
</tr>
</tbody>
</table>
Starting sybcluster  The recommended method for starting sybcluster and connecting to a cluster is:

```bash
sybcluster -U login_name -P password -C cluster_name
-P agent_spec
```

**Example 1**  Starts sybcluster using direct connect and default port numbers.

```bash
sybcluster -U uafadmin -P -C mycluster
-P "blade1,blade2,blade3"
```

**Example 2**  Starts sybcluster using direct connect and port numbers.

```bash
sybcluster -U uafadmin -P -C mycluster
-P "blade1:9100,blade2:9292,blade3:9393"
```

**Example 3**  You can also start sybcluster using discovery. See “Discovery methods” on page 282.

```bash
sybcluster -U uafadmin -P -C mycluster
-d "JINI(myjiniserver:4564)"
```

The `-C cluster_name`, `-P password`, `-I instance_name`, `-F agent_connection`, and `-d discovery_list` parameters are default values that can be changed using subsequent sybcluster interactive commands. If you do not specify these values on the sybcluster command line, sybcluster prompts for them as they are required.

You can also start sybcluster and then use the interactive connect command to connect to the cluster. For example:

```bash
sybcluster
> connect to mycluster login uafadmin password " "
agent "blade1,blade2,blade3"
```

**Note**  See “sybcluster interactive commands” on page 65 for syntax and usage descriptions of the sybcluster interactive commands.

---

**Parameters**

- `-C cluster_name`
  
  is the unique name of the Sybase shared-disk cluster to be managed. sybcluster looks up the name in the cluster directory or uses agent discovery services.

- `-d discovery_list`
  
  specifies the discovery services to be used to discover a shared-disk cluster agent and the discovery order. Discovery services supported for the Cluster Edition are listed in Table 8-1. The format is:
"method([(method_specification) [...])]"

For example:

- `d "udp(),jini(jinihost1;jinihost2)"

### Table 8-1: Discovery methods

<table>
<thead>
<tr>
<th>Discovery method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP()</td>
<td>Performs a UDP broadcast and listens for a response from listening Unified Agents. UDP discovery does not cross subnet boundaries.</td>
</tr>
<tr>
<td>JINI(JINI_spec)</td>
<td>Specifies the JINI servers used to look up the locations of nodes in the cluster. The specification form is: <code>host-name[:port_num]</code>. Indicate multiple JINI servers by placing a semicolon between each specification. By default, <code>sybcluster</code> uses port number 4160 to attach to a JINI server. The JINI server must be running, and the management agents (UAF) must be registered with the JINI server. The locations of the nodes, and status of the instances are stored on the JINI server.</td>
</tr>
</tbody>
</table>
| LDAP(LDAP_spec)  | Specifies an LDAP server that will be used to look up the locations of the nodes in the cluster. The specification form is: `host_name[:port_num][?registry]`. Indicate multiple LDAP servers by placing a semicolon between each specification. By default, `sybcluster` uses port number 389 to attach to an LDAP server and the LDAP directory at “cn=ua-registry,ou=ua,dc=sybase,dc=com”.

- `-F agent_connection`
  specifies the agent to be used to access the cluster. The format is:

  `host_name[:port_num] [, host_name[:port_num]]`

  For example:

  `-F "node1,node2,node3,node4:9999"

  The default port number is 9999.

- `-h`
  displays `sybcluster` syntax and lists supported interactive commands.

- `-I instance_name`
  specifies the instance to be accessed. If you do not specify the `-I` option when you execute `sybcluster`, you may need to specify it when entering certain interactive commands. `sybcluster` uses this name to discover the remote host, and as a default when executing interactive commands. If an interactive command affects multiple instances, the instance identified by `-I`, if available, is used as the priority connection. To override the instance specified by `-I`, execute the `use` command in interactive mode.
-i
specifies an operating system file for input to sybcluster. This file contains sybcluster commands, one command per line. The final command in the file should be quit.

-L
creates a sybcluster.log file. sybcluster writes all messages to this file irrespective of the message level set by the -m option.

-m message_level
specifies which sybcluster and unified agent messages are displayed on the client console. Message levels are:

- 0 – off (no messages to log file or console)
- 1 – fatal
- 2 – error
- 3 – warning
- 4 – information
- 5 – debug

sybcluster displays all messages of the level you choose and all messages of greater severity (with lower numbers). That is, if you select message level 3, sybcluster displays messages of level 3, 2, and 1. The default level is 4.

-P [password]
is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password. This is the Simple Login Module in the Agent configuration. The user and password can be configured to use several different mechanisms for authentication and authorization, including using the running instance and the operating system logins.

If you do not specify the -P option, sybcluster prompts for a password. For a blank or null password, use the -P option without a value or enter a set of quotation marks without content.

You can encrypt the password using the Sybase passencrypt utility. See the Clusters Users Guide.

-U user_name
is the management agent login name. The default login after installation is “uafadmin.” See the -P description for more information.
-v

displays the sybcluster version number and other information.

**sybcluster interactive commands**

This section describes the sybcluster interactive commands. Some commands are active before you connect to a cluster (see “Commands active before connecting to a cluster” on page 284); others are active only after you connect to a cluster (see “Commands active after connecting to a cluster” on page 285).

The sybcluster command prompt includes the current cluster and the default instance when these values have been set. The prompt is:

- `>` – when sybcluster is not connected to a cluster.
- `cluster_name>` – when sybcluster is connected to a cluster.
- `cluster_name instance_name>` – when sybcluster is connected to a cluster and a default instance has been set.

These commands are active before you connect to a cluster. They are not available after you connect to a cluster.

**Connect**

```
[ to cluster_name ]
[ Login login_name ]
[ Password [password] ]
[ Agent agent_spec ]
[ Discovery discovery_spec ]
```

**Create Cluster**

```
[ cluster_name ]
[ Login login_name ]
[ Password password ]
[ Agent agent_spec ]
[ Discovery discovery_spec ]
[ File input_file ]
```

**Deploy Plugin**

```
[ Login login_name ]
[ Password password ]
[ Agent agent_spec ]
[ Discovery discovery_spec ]
```

**Exit**

Help

**Quit**

Show agents

```
[ login login_name ]
[ password password ]
```
Utility Guide

CHAPTER 8  Using sybcluster to Manage the Cluster

[ agent "agent_spec[, agent_spec[,...]]" ]
[ discovery "discovery_spec[, discovery_spec[,...]]" ]

Upgrade server server_name
[ login login_name ]
[ password password ]
[ agent agent_spec ]
[ discovery discovery_spec ]
[ file input_file_name ]
[ checkonly ]

Commands active after connecting to a cluster

These commands are active only after you connect to a cluster:

Add
Instance instance_name
file file_name
Backupserver

Create
Backupserver
Monitorserver
Xpserver

Disconnect

Diagnose
Cluster
Instance instance_name

Drop
Backupserver
Cluster
Instance instance_name
Monitorserver
Xpserver

Localize

Set
Cluster
MaxInst max_num_instances
Login [ login_name ] [ Password password ]
TraceFlags trace_flag
Primary Protocol protocol
Secondary Protocol protocol
Instance [ instance_name ]
LogPath log_file_path
Primary
Address ip_address
Port port_range_start port_range_end
Secondary
Address ip_address
Port port_range_start port_range_end
StartArgs startup_arguments
BackupServer
MonitorServer
XPServer
Port

Show
Cluster
  Config
  Template
  Log
    [ Errors]
    [ MinSeverity severity_level ]
    [ StartDate [date_string]]
    [ EndDate [date_string]]
    [ Last num_of_lines ]
Status

Instance [instance_name]
  Config
  Log
    [ Errors ]
    [ MinSeverity severity_level ]
    [ StartDate [date_string]]
    [ EndDate [date_string]]
    [ Last num_of_lines ]
Status

BackupServer
  Config
  Membership mode

MonitorServer
  Config
  Session

XPServer
  Config

Shutdown
Cluster
  Instance [ instance_name ]

Start
Cluster
  Instance instance_name
    [ Unlock ]

Use [ instance_name ]
CHAPTER 8  Using sybcluster to Manage the Cluster

add backupserver

Description  Configures Backup Server for nodes not already configured for Backup Server.
Syntax      add backupserver
Examples    Adds a Backup Server to “mycluster” on nodes “blade3” and “blade4”.
            add backupserver
Finding nodes for which Backup Server is not configured...
Do you want to configure Backup Server for node "blade3"? [Y]
Please enter the Backup Server port number for node "blade3": 5001
Do you want to configure Backup Server for node "blade4"? [Y]
Please enter the Backup Server port number for node "blade4": 50011

Usage
• You can configure Backup Server for one or more nodes in the cluster.
• add backupserver lets you add additional nodes when configuring for single Backup Servers. You cannot use this command to add multiple Backup Servers.

add instance

Description  Adds one new instance to the cluster. The instance can be added interactively, with sybcluster prompting for necessary configuration information, or through an input file. add instance also creates a local system temporary database for the new instance. add instance prompts vary depending on whether configuration for the cluster is shared or private.
Syntax      add instance instance_name [ file "input_file" ]
add instance

Parameters

*instance_name*

is the name of the instance.

*file "input_file"*

specifies a file name that contains the configuration information for adding an instance.

Usage

- add instance creates a local system temporary database for the new instance. Before executing add instance, make sure that a device with sufficient space for the local system database exists.

- The input file for add instance has the same format as the cluster input file. However, the add instance input file may limit the instance definitions to the new instance in the node section.

- add instance may prompt for this information:
  - The instance name, if you did not enter an instance name in the command statement.
  - The node hosting the instance
  - The port number of the UAF agent on the node
  - The query port number
  - The primary and secondary address of the node
  - The primary and secondary port specification

- If you have configured single Backup Servers for the cluster, add instance asks whether Backup Server is already configured for the new instance node. If no, add instance asks if Backup Server should be configured. If yes, it prompts for the Backup Server port for the node.

If you have configured multiple Backup Servers for the cluster, add instance prompts for:

  - The name of the Backup Server
  - The Backup Server log file path
  - The Backup Server port for the new instance

add instance also prompts for Monitor Server and XP Server port number information for the new instance.

- If the installation mode is private, add instance prompts for additional information for the new instance:
  - The $SYBASE home directory
• The environment shell script path
• The Adaptive Server home directory
• The server configuration file path
• The interfaces file path, if LDAP is not configured
connect

Description
Connects to an existing cluster.

Syntax
connect [ to cluster_name ]
[ login login_name ]
[ password [password ] ]
[ agent "agent_spec [, agent_spec [...]]" ]
[ discovery "discovery_spec [, discovery_spec [...]]" ]

Parameters

cluster_name
is the name of the cluster to which you are connecting.

login login_name
is the management agent login for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

For information about Sybase Common Security, see Unified Agent and Agent Management Console Version 2.0 for Windows and UNIX.

password password
is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

agent agent_spec
is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is node_name:port_number[, node_name:port_number ][,...]. The default port number is 9999.

This is the preferred method for connecting to a cluster.

discovery discovery_spec
is the discovery method used to identify the agents responsible for the requested cluster. The format is method[(method_specification)][, (method_specification )][,...]. See Table 8-1 on page 282.

Examples

Example 1 Connects to “mycluster,” when “mycluster” is the default cluster specified in the sybcluster command statement.
connect

**Example 2** In this example, you connect to “mycluster” using the agent specification and default port numbers.

```
connect to mycluster agent "blade1,blade2,blade3"
```

**Usage**

- A direct connection is one in which the user identifies the cluster nodes and, optionally, the port numbers for the UAF agents. Sample agent specifications are:
  - `myhost` – identifies the host node and assumes the default listening port of 9999.
  - `myhost.mydomain.com` – includes the host domain name.
  - `myhost:9999` – identifies the host node and listening port number.

---

**create backupserver**

**Description**

Creates a Backup Server for the cluster, or, if the cluster has been configured for multiple Backup Servers, creates a Backup Server for each instance in the cluster.

**Syntax**

```
create backupserver
```

**Examples**

**Example 1** Creates the Backup Server “mycluster_BS” for “mycluster”:

```
create backupserver
Do you want to create multiple Backup Servers? [Y] N
Enter the Backup Server name: [mycluster_BS]
Enter the Backup Server log file path: [$SYBASE/ASE-15_0/install/mycluster_BS.log]
Do you want to create a Backup Server for node "blade1"? [Y]
Enter the Backup Server port number for node "blade1":
The Backup Server "mycluster_BS" was successfully defined.
```

**Example 2** Creates multiple Backup Servers for “mycluster” running on “ase1” on “blade1” and “ase2” on “blade2.”

```
create backupserver
Do you want to create multiple Backup Servers? [Y] Y
The "dump/load" commands would be routed to appropriate Backup Server based on following policies:
1. Dedicated - Each instance associated with exactly one Backup Server.
2. Round Robin - Choose the Backup Server with least number of requests in
```
create cluster

round robin fashion starting from global cluster level counter. Enter the number corresponding to the policy to be used: [1] 1

Enter the Backup Server name for instance ase1: [ase1_BS] Enter Backup Server log file path: [/remote/var/sybase/install/ase1_BS_log] Enter the Backup Server port for node "blade1": 23001

Enter the Backup Server name for instance ase2: [ase2_BS] Enter Backup Server log file path: [/remote/var/sybase/install/ase2_BS_log] Enter the Backup Server port for node "blade2": 23002

Backup Servers successfully defined.

Usage

- create backupserver prompts for the Backup Server listening port on each node. It copies other necessary configuration information from the cluster configuration file. create backupserver:
  - Creates directory service entries for Backup Server on each node.
  - Creates the Backup Server configuration and log files, and the RUN_<backup_server> script.
  - Adds the Backup Server name to the cluster’s sys.servers table.
  - Enables Backup Server HA.

create cluster

Description

Creates an Adaptive Server shared-disk cluster. Enter the necessary configuration information interactively, as responses to a series of prompts, or use an input file.

Syntax

create cluster [cluster_name ]
[ login login_name ]
[ password password ]
[ agent "agent_spec [, agent_spec [ , ... ]]
[ discovery "discovery_spec [, discovery_spec [ , ... ]]
[ file "input_file"]

Parameters

cluster_name

is the name of the cluster.
CHAPTER 8   Using sybcluster to Manage the Cluster

login login_name

is the management agent login for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the simple login module in the agent configuration. You can configure the user name and password to use several different mechanisms for authentication and authorization, including operating system logins.

password password

is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. You can configure the user name and password to use several different mechanisms for authentication and authorization, including operating system logins.

agent agent_spec

is the agent specification that identifies the nodes in the cluster running a Unified Agent, and the port number that sybcluster uses to connect to the Unified Agent. The format is node_name:port_number [, node_name:port_number ] [...]. The default port number is “9999.”

discovery discovery_spec

is the discovery method used to identify the agents responsible for the requested cluster. The format is method[(method_specification)] [, (method_specification) [...]]. See Table 8-1 on page 282.

file “input_file”

is the operating system input file for creating the cluster.

Examples

Example 1 Creates a new cluster called “mycluster”; sybcluster prompts you for the information necessary to create the cluster.

create cluster mycluster

Example 2 Creates a new cluster called “mycluster1” using configuration information supplied in the mycluster1.xml file.

create cluster mycluster1 file mycluster1.xml

Usage

• When you create a cluster, sybcluster prompts for the:
  • Cluster name, if one has not been provided  
  • Number of instances  
  • Installation mode for the cluster (private or shared)
create monitorserver

- Complete path to the master, quorum, PCI, systemdb, sybsysprocs, and temporary database devices
- Path to the interfaces file, if LDAP is not configured and this is a shared install
- Trace flags (optional)
- Complete path to the dataserver configuration file, if this is a shared install
- Primary and secondary interconnection protocols
- Instance host name, port number, private address, log file location, and start-up arguments.

If this is a private install, sybcluster also prompts for the $SYBASE installation directory, Adaptive Server home directory, dataserver configuration file location, and interfaces file location (if LDAP is not configured).

- If sybcluster detects the Veritas Cluster Server (VCS) on the system, sybcluster asks if it should check whether device is managed by VCS.
- After you create and confirm the cluster, create cluster prompts for an I/O fencing check, which checks whether or not each device has I/O fencing capability (see the Installation Guide for a description of I/O fencing).

create monitorserver

Description Creates a Monitor Server for each instance in the cluster.

Note You must run the installmon isql script, located in $SYBASE:$SYBASE_ASE/scripts, before running Monitor Server.

Syntax create monitorserver

Examples Create a Monitor Server for each instance in "mycluster".

create monitorserver

Enter the Monitor Server log file location for instance "ase1". [$SYBASE/ASE-15_0/install/mycluster_MS1.log]
Enter the Monitor Server port number for instance "ase1": Enter the Monitor Server log file location for instance "ase2": [
CHAPTER 8 Using sybcluster to Manage the Cluster

ASE-15_0/install/mycluster_MS2.log
Enter the Monitor Server port number for instance "ase2":
Enter the Monitor Server log file location for instance "ase3".[$SYBASE/
ASE-15_0/install/mycluster_MS3.log]
Enter the Monitor Server port number for instance "ase3":
Enter a username: sa
Enter a password:
WARN — mycluster:AseCfg:411: The SQL script "INSTALLMON" must be run before
the Monitor Server can be run.
The Monitor Server has been successfully configured.

Usage create monitorserver prompts for the listening port number for each instance in
the cluster, and a user name and password.

create xpserver
Description Creates an XP Server for each instance in the cluster.
Syntax create xpserver
Examples Creates an XP Server for each instance in "mycluster".

create xpserver

Enter the XP Server port number for instance "ase1":
Enter the XP Server port number for instance "ase2":
Enter the XP Server port number for instance "ase3":
The XP Server was successfully defined for each instance.

Usage create xpserver prompts for the XP Server listening port for each node in the
cluster. Other information necessary to create the XP Server is read from the
cluster configuration file.

deploy plugin
Description Adds the configuration information for a single instance of the cluster to the
Unified Agent. Can be used to configure the Unified Agent to manage a cluster
if you created the cluster without using the Adaptive Server plug-in or
sybcluster utility, or if you need to recreate the Unified Agent configuration for
a cluster. The configuration of a cluster instance is performed by deploying a
Unified Agent plug-in.
deploy plugin

Syntax

deploy plugin
  [ login login_name ]
  [ password password ]
  [ agent agent_spec ]
  [ discovery discovery_spec ]

Parameters

login login_name
  is the management agent login for the Sybase Common Security
  Infrastructure in the Unified Agent framework. The default user name after
  installation is “uafadmin” with no password; this is the Simple Login
  Module in the Agent configuration. The user name and password can be
  configured to use several different mechanisms for authentication and
  authorization, including operating system logins.

password password
  is the management agent password for the Sybase Common Security
  Infrastructure in the Unified Agent framework. The default user name after
  installation is “uafadmin” with no password; this is the Simple Login
  Module in the Agent configuration. The user name and password can be
  configured to use several different mechanisms for authentication and
  authorization, including operating system logins.

agent agent_spec
  is the agent specification that identifies the nodes in the cluster running a
  Unified Agent, and the port number that sybcluster uses to connect to the
  Unified Agent. The format is “node_name:port_number [, node_name:port_number ] [...]]”. The default port number is “9999”.

discovery discovery_spec
  is the discovery method used to identify the agents responsible for the
  requested cluster. The format is “method[(method_specification)] [, ( method_specification ) [...]]”. See Table 8-1 for more information about
  discovery methods.

Examples

Example 1 Deploys the plug-in using the uaf agent on host “system1501”.

    deploy plugin agent system1501

sybcluster prompts for the cluster name, cluster node number, installation
mode, full path to the quorum device, the environment shell script path, and the
Adaptive Server home directory.

Example 2 Deploys the plug-in and uses discovery to identify the agent:

    deploy plugin discovery udp

Usage

• After you execute deploy plugin, sybcluster prompts you for the:
  • Path to the quorum device.
• Path to the Sybase home directory.
• Installation mode (private or shared). The default is shared.
• The location of your Sybase environment script. This must be a shell script that can be loaded using the “file_name” syntax such as “sh” or “bash”. An example is SYBASE.sh.
• The location of your Adaptive Server software directory. The default is <sybase_home_directory>/ASE-15_0. When entering the location of the Adaptive Server software directory, include the full path. Do not use $SYBASE.
• The dataserver login and password are configured using the login command, which updates all Adaptive Server plug-ins managing the cluster.

diagnose cluster

Description
Performs a set of checks to ensure that the cluster is working correctly.

Syntax
diagnose cluster

Examples
Checks that “mycluster” is working correctly.

diagnose cluster
Cluster name..................mycluster
Maximum instances............4
Cluster node count..........1
Instances defined............4
Is cluster locked...........Yes
JDBC connection available....1 ase1 Yes
JDBC connection available....2 ase2 Yes
JDBC connection available....3 ase3 Yes
JDBC connection available....4 ase4 Yes
Instance Public Network.....1 ase1 on blade1 (10.22.79.39) Reachable: Yes
Instance Public Network.....2 ase2 on blade1 (10.22.79.39) Reachable: Yes
Instance Public Network.....3 ase3 on blade1 (10.22.79.39) Reachable: Yes
Instance Public Network.....4 ase4 on blade1 (10.22.79.39) Reachable: Yes
Has private Primary network: No
Has private Secondary network: No
Network ports required/instance 20
Minimum port allowed........1025
Maximum port allowed........65535
diagnose instance

Current port strategy............ Public primary and secondary unique.
...The ports are sequenced primary followed by the next instance primary.
...When the primaries are completed the secondary ports follow the same
pattern.

Recommended port strategy....... Public primary and secondary unique.
...The ports are sequenced primary followed by the next instance primary.
...When the primaries are completed the secondary ports follow the same
pattern.

Usage

- diagnose cluster checks that:
  - A Unified Agent is running on each instance in the cluster.
  - The number of instances in the cluster does not exceed the value set
    for maximum number of instances.
  - The quorum file exists.
  - All instances are defined in the interfaces file and that port numbers
don’t conflict.
  - The primary and secondary protocol specifications do not overlap.
  - Each of the $SYBASE directories are shared.

diagnose instance

Description
Performs a set of checks to ensure that the instance is configured correctly.

Syntax
diagnose instance [instance_name]

Parameters
instance_name
is the name of an instance. If no instance name is entered, sybcluster uses
the default value.

Examples
Displays and verifies configuration information for “ase1” on “mycluster”.

diagnose instance ase1
Cluster name ................. mycluster
Instance id .................. 1
Instance name ................. ase1
Node name ..................... blade1
Query port ................... 7101
JDBC connection available .... Yes

Instance Public Network ....... 1 asel on blade1 (10.33.108.139)
Reachable:......Yes

Minimum port allowed.......... 1025
Maximum port allowed .......... 65535

Instance port range ........... 1 Primary asel 17100 to 17115 (16) Okay
Instance port range ........... 1 Secondary asel 17165 to 17180 (16) Okay

Usage
Use diagnose cluster to ensure the cluster is configured correctly.

disconnect
Description
Closes all connections to the current cluster and returns sybcluster to an unconnected state.

Syntax
disconnect

Usage
Use connect to reconnect to an existing cluster.

drop backupserver
Description
Drops Backup Server from a node or from the cluster. If the cluster has been configured for multiple Backup Servers, drops all Backup Servers.

Syntax
drop backupserver

Examples
Example 1 Drops a single Backup Server.

drop backupserver
Do you want to drop the Backup Server from:
1. Selected nodes
2. Cluster
Enter choice: 1
Do you want to drop Backup Server from node "blade1"? [N] y
Do you want to drop Backup Server from node "blade2"? [N]
The Backup Server has been dropped from selected nodes.
**drop cluster**

**Example 2** Drops the Backup Server from the cluster.

drop backupserver

Do you want to drop the Backup Server from:
  1. Selected nodes
  2. Cluster
Enter choice: 2
Are you sure you want to drop Backup Server mycluster_BS from cluster mycluster? (Y or N): [N] y
The Backup Server has been dropped.

**Example 3** Drops all Backup Servers when multiple Backup Servers have been configured for the cluster:

drop backupserver

Multiple Backup Server are defined for the cluster. This command will drop all of them.
Are you sure you want to continue? (Y/N): [N] y
The Backup Server has been dropped.

**Usage**

Use `drop backupserver` to drop a Backup Server from the cluster.

---

**drop cluster**

**Description**

Removes each instance from a cluster and then removes the cluster definition from the cluster configuration file. Also, removes regular files associated with the cluster and the cluster agent plug-ins that manage the cluster. The cluster must be Down to use `drop cluster`.

**Syntax**

drop cluster

**Examples**

Drops all instances from the current cluster and deletes the cluster.

drop cluster

**Usage**

- `sybcluster` prompts for confirmation before dropping the cluster.
- `drop cluster`:
  - Removes cluster and instance entries from the interfaces file, configuration files, and specified data devices.
  - Marks the quorum device as unused.
  - Shuts down and removes the cluster’s UAF agent plug-ins.
Due to certain file system locking, the UAF plug-ins may not be deleted after you use `drop cluster`. Verify that the `SYBASE_UA/nodes/*/plugins/<cluster_name>` directory has been deleted. If the directory still exists, delete it.

### drop instance

**Description**
Removes an instance from the cluster configuration file and updates the Unified Agent Framework (UAF) and discovery services. Also, notifies the cluster that an instance is to be dropped, and removes the instance and interfaces file entries.

**Syntax**
```
drop instance [ instance_name ]
```

**Parameters**
- `instance_name` identifies an instance in a cluster. If an instance name is not specified, `sybcluster` uses the default specified in the `sybcluster` command line.

**Examples**
Removes the “ase3” instance from the current cluster.
```
drop instance ase3
```
**drop monitorserver**

**Description**
Drops the Monitor Server for every instance in the cluster.

**Syntax**
drop monitorserver

**Examples**
Drops all Monitor Servers defined for “mycluster”.

drop monitorserver

Are you sure you want to drop the Monitor Server configurations from the cluster mycluster? (Y or N): [N] y

The Monitor Servers have been dropped for all instances.

**Usage**
Use drop monitorserver to drop a Monitor Server from the cluster.

**drop xpserver**

**Description**
Drops the XP Server for each instance in the cluster.

**Usage**
CHAPTER 8    Using sybcluster to Manage the Cluster

Syntax  drop xpserver
Examples  Drops the XP Servers for “mycluster”.

    drop xpserver
    Are you sure you want to drop the XP Servers from cluster
    mycluster”? {Y or N}: [N] y
    The XP Servers have been dropped for all instances.

Usage  Use drop xpserver to drop an XP Server from the cluster.

exit
Description  Exits the sybcluster utility.
Syntax  exit
Usage  • exit and quit perform the same task: they exit the sybcluster utility.
        • If some agents have been shut down while connected to sybcluster, error
          messages describing the connections may display. You can ignore these
          messages.

help
Description  Lists the currently available sybcluster interactive commands.
Syntax  help
Usage  The list of currently available interactive commands changes depending on
        whether or not sybcluster is connected to a cluster.

localize
Description  Displays the current values for default language, charset, and sort order.
             Allows modification of default values, and addition or removal of languages.
Syntax  localize
Examples

This example displays default localization values, and then prompts for changes. The default character set changes to Chinese, the default charset to eucgb, and the default sort order to bin_eucgb.

```
localize
Current default locale properties are:
Default Language - portuguese
Default Charset - mac
Default SortOrder - Binary ordering, for use with the Macintosh character set (mac).

Options for default Language are:
1. spanish
2. portuguese
3. german
4. us_english
5. thai
6. french
7. japanese
8. chinese
9. korean
10. polish
Enter the number representing the language to be set as defaults: [2] 8

Options for default charsets are:
1. gb18030
2. eucgb
3. utf8
Enter the number representing the charset to be set as default: [1] 2

Options for sort orders are:
1. Binary ordering, for the EUC GB2312-80 character set (eucgb).
Enter the number representing the sort order to be set as default [1]

Do you want to install any language? [Y] n
Do you want to remove any language? [N]
The cluster mycluster was successfully localized with default language chinese, charset eucgb, sortorder bin_eucgb
```

Usage

- The current default localization value displays after each prompt. To accept the current value, enter a carriage return instead of a number.
- The options for default languages include all languages present in $SYBASE_ASE. If the selected default language is not configured, use localize to configure it or remove it.
• To ensure that new values are consistent for all instances in the cluster, restart the cluster after changing localization values.

**quit**

Description: Exits the sybcluster utility.

Syntax: quit

Usage: exit and quit both exit the sybcluster utility.

**set backupserver**

Description: Changes the listening port number for Backup Server on specified nodes in a cluster.

Syntax: set backupserver

Examples

**Example 1** Changes the listening port number for Backup Server on “blade1” of “mycluster”.

```
set backupserver
Backup Server is configured on the following nodes:
  1. blade1: 3001
  2. blade2: 3002
  3. blade3: 3003
Do you want to change the Backup Server port on any node? {Y}
Enter the number representing the node whose port you want to change: 1
Enter the Backup Server port number for node "blade1": 4001
Backup Server was successfully modified as per new properties.
```

**Example 2** When “mycluster” has been configured for multiple Backup Servers, changes the listening port number for one or more Backup Servers.

```
set backupserver

Multiple Backup Servers are configured for the cluster. Their configuration is as follows:
Backup Server Policy: Dedicated
  1. Backup Server name: ase1_BS
```
set cluster

Configured for blade1:23001
Log file location: /remote/sybase/ASE-15_0/install/ase1_BS.log
2. Backup Server name: ase2_BS
    Configured for blade2:23002
    Log file location: /remote/sybase/ASE-15_0/install/ase2_BS.log
3. Backup Server name: ase3_BS
    Configured for blade3:23003
    Log file location: /remote/sybase/ASE-15_0/install/ase3_BS.log

Do you want to edit any Backup Server (y/n)? [Y]
Enter the number representing the Backup Server you want to edit: [1]
Enter the new port for Backup Server "ase1_BS": 24001
Do you want to edit any more Backup Servers? [N]

Backup Server "ase1_BS" successfully updated.

Usage
When you set a new listening port number, Adaptive Server first checks to see if that port number is already in use.

set cluster

Description
Changes configuration values for the cluster. The cluster must be down to execute all set cluster commands except set cluster login.

Syntax
set cluster { maxinst max_num_instances | traceflags trace_flag[ trace_flag[...]] | (primary | secondary ) protocol udp | login login_name [password password ] }

Parameters
maxinst max_instances
specifies the maximum number of instances that can run in the cluster.

traceflags trace_flag[ trace_flag[...]]
specifies trace flags to be set when the cluster starts.
CHAPTER 8 Using sybcluster to Manage the Cluster

login login_name [ password password ]
specifies a user name and password that the Unified Agent uses to log in to the cluster and perform shutdown and certain other tasks. This login must have sa_role. By default, the Unified Agent uses the "sa" login with no password. To change this password, use set cluster login. See the Clusters Users Guide.

Note set cluster login can only be used to change the login or password that the Unified Agent uses to log in to the cluster. To change the login or password sybcluster uses to log in to the Unified Agent, use the Agent Management Console Sybase Central plug-in.

{ primary | secondary } protocol udp
sets the protocol for the private network for the primary or secondary interface.

Examples

Example 1 Changes the maximum number of instances to 4 for “mycluster”.

    set cluster maxinst 4

Example 2 Adds the trace flag 15506.

    set cluster traceflags 15506

Example 3 Changes the password for the “sa” user name.

    set cluster login sa password abcde

Usage
To check that the cluster is Down, enter show cluster status.

set instance

Description Sets properties of the instance. The instance must be Down.

Syntax set instance instance_name logpath path
set instance instance_name startargs values
set instance instance_name { primary | secondary } port port_range
set instance instance_name { primary | secondary } address ip_address

Parameters logpath logfile_path
    specifies the path for the instance log file.

    instance_name
    specifies an instance.
**set monitorserver**

**Description**
Changes the Monitor Server listening port numbers for specified instances in the cluster.

**Syntax**
```
set monitorserver
```

**Examples**
Changes the Monitor Server listening port number for instance “ase2” on “blade2” of “mycluster” without changing the listening port numbers for other instances in the cluster.

```
set monitorserver
Enter the Monitor Server Port number for instance "blade1": [5001] <CR>
Enter the Monitor Server port number for instance "blade2": [5002] 5011
Enter the Monitor Server port number for instance "blade3": [5003] <CR>
```

**Usage**
- set monitorserver prompts you to change the listening port number for each Monitor Server configured in the cluster. You can accept the current value, provided in square brackets, or enter a new port number.
- You can change the Monitor Server listening port number for one or more instances.

**setargs startup_args**

specifies arguments for starting the instance.

**primary | secondary** address ip_address
specifies the primary or secondary IP address for the instance.

**primary | secondary** port port_range
specifies the primary or secondary port range for the instance. The format for port_range is : start_num end_num.

**Examples**
Changes the port range for the primary interface listening port.
```
set instance primary port 7777
```

**Usage**
To check that the instance is Down, enter show cluster status.
**set xpserver port**

**Description**
Changes the listening port number for XP Server on specified nodes of the cluster.

**Syntax**
`set xpserver port`

**Examples**
Changes the listening port for the XP Server for instance “ase1” on “blade1” of “mycluster” without changing the listening ports for “ase2” and “ase3”.

```
set xpserver port
Enter the XP Server port number for instance "ase1" [3002]: 4002
Enter the XP Server port number for instance "ase2" [3002]: <CR>
Enter the XP Server port number for instance "ase3" [3002]: <CR>
```

**Usage**
You can change the XP Server listening port number on one or more instances.

**show agents**

**Description**
Displays information about available UAF agents.

**Syntax**
`Show agents`

```
[ login login_name ]
[ password password ]
[ agent "agent_spec[, agent_spec[,...]]" ]
[ discovery "discovery_spec[, discovery_spec[,...]]" ]
```

**Parameters**
- `login login_name`
  is the management agent login for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

- `password password`
  is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.
show agents

agent agent_spec
is the agent specification that identifies the nodes in the cluster running a
Unified Agent, and the port number that sybcluster uses to connect to the
Unified Agent. The format is "node_name:port_number[, 
node_name:port_number][,...]". The default port number is “9999.”

discovery discovery_spec
is the discovery method used to identify the agents responsible for the
requested cluster. The format is "method[(method_specification)] [, ( 
method_specification) [...]]". See Table 8-1 for more information about
discovery methods.

Examples
Displays UAF agent information:

    show agents
Agent Information: service:jmx:rmi:///jndi/rmi:///blade1:9985/agent
-----------------------------------------------------
Node Name: blade1
Agent Port: 9985
Agent Version: 2.5.0
Agent Build: 977
OS Name: Linux
OS Version: 2.6.9-42.ELsmp
OS Architecture: amd64
Agent Service Info:
Agent Service (Agent) Build: 977 Status: running
BootstrapService (BootstrapService) Build: <unavailable> Status: running
Configuration Service (ConfigService) Build: 977 Status: running
Deployment Service (DeploymentService) Build: <unavailable> Status: running
Environment Service (EnvironmentDiscoveryService) Build: 977 Status: running
File Transfer Service (FileTransferService) Build: 977 Status: running
Plugin Registration Service (PluginRegisterService) Build: 977 Status: running
RMI Service (RMIService) Build: 977 Status: running
Remote Shell Service (RemoteShellService) Build: 977 Status: running Security
Service (SecurityService) Build: 977 Status: running Self Discovery Service
(SelfDiscoveryService) Build: 977 Status: running Service Registration Service
(ServiceRegistrationService) Build: 977 Status: running Session Service
(SessionService) Build: 977 Status: running Sybase Home Service (SybaseHomeService)
Build: 14 Status: running
Agent Plugin Info:
ASE Cluster Agent Plugin (com.sybase.ase.cluster) Version: 15.1.0 Build: 85 Instance:
1 Status: running
Cluster Name: marion
CHAPTER 8   Using sybcluster to Manage the Cluster

Env Shell: /job1/miso/betaR1/SYBASE.sh  Shell Type: sh
Sybase Home: /job1/miso/betaR1
ASE Home: /job1/miso/betaR1/ASE-15_0
ASE Version: Adaptive Server Enterprise/15.0.1/EBF 14721 Cluster
ASE Login: sa
Update Time: 60 seconds
Last Update: 2007-09-28 22:09:02 -0700

Usage

show agents is active before you connect to a cluster.

show backupserver config

Description
Displays the nodes on which Backup Server is configured, the associated
listening port numbers, and the Backup Server policy.

Syntax
show backupserver config

Examples
Displays configuration information for “mycluster”, which has been
configured for multiple Backup Servers.

Usage

• Use the show backupserver config command to display Backup Server
  configuration information.

• If you are configuring multiple Backup Servers, show backupserver config
  includes the Backup Server policy.
show cluster

Description Displays configuration, log, and status information about the cluster.

Syntax

```
Show
    Cluster
    Config
    Template
    Log
    [Errors]
    [MinSeverity severity_level]
    [StartDate [date_string]]
    [EndDate [date_string]]
    [Last number_of_lines]
    Status
```

Parameters

status displays status information for the cluster. Values are:

- Up
- Down
- Undefined
- Invalid
- Start
- Init
- Quiesce

log displays logs from all instances in the cluster.

errors [ minseverity severity_level ] display log file entries for errors. Optionally, limit displayed error entries to a severity level and above.

Note Error severities_level is an attribute of Adaptive Server error messages, not sybcluster messages.

startdate [ date_string ] display log file entries that occur on and after the date specified. The format for date_string is: mm:dd:yy.

If you do not specify a startdate or enddate date_string, the default is the current date (today).
**enddate** [ *date_string* ]

display log file entries that occur on or before the date specified.

**last** *num_lines*

Limits the number of lines displayed, counting backward from the last line in the log files.

**config**

displays configuration information for the cluster:

- Maximum number of instances
- Installation mode: shared or private
- Primary and secondary protocols
- Trace flags set
- Location and name of the quorum device
- LDAP information, if LDAP is configured
- Location and name of the master device

**template**

displays formatted configuration information for the cluster.

**Examples**

**Example 1** Displays current configuration and other information about the default cluster.

```
show cluster status
```

```
<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Node</th>
<th>State</th>
<th>Heartbeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ase1</td>
<td>blade1</td>
<td>Up</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>ase2</td>
<td>blade2</td>
<td>Up</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>ase3</td>
<td>blade3</td>
<td>Down</td>
<td>No</td>
</tr>
</tbody>
</table>
```

**Example 2** Displays configuration information (including LDAP, if it is configured) for the default cluster configured for shared installation mode.

```
show cluster config
```

```
**Cluster configuration for "mycluster"**
Installation Mode shared
Interfaces Path "/work2/sybase/ASE-15_0/"
Trace Flags: 15556
Maximum Instances "4"
Quorum "/dev/raw/raw101"
```
**Example 3** Displays configuration information for the default cluster configured for private installation mode.

```plaintext
show cluster config

**Cluster configuration for "localcluster" **
Installation Mode "private"
Trace Flags:
There are no trace flags
Maximum Instances "4"
Quorum "/dev/raw/raw101"
Master Device "/dev/raw/raw102"
logfile ase1 /remote/work2/sybase/ASE-15_0/install/ase1.log
run_parameters ase1 null
logfile ase2 /work2/sybase/ASE-15_0/install/ase2.log
run_parameters ase2 null
Primary Interconnect "udp"
  Server[1] ase1 tigger.sybase.com 26016 26031
Secondary Interconnect "udp"
  Server[1] ase1 tigger.sybase.com 26081 26096
LDAP server blade1 2250
```

Usage

```
show cluster status displays the results of a show instance command on each instance in the cluster.
```
show instance

Description
Displays information about an instance.

Syntax
show instance [instance_name] {
  config |
  status |
  log
    [ [ errors ] minseverity severity_level ] |
    [ startdate [ date_string ] ] |
    [ enddate [ date_string ] ] |
    [ last num_lines ] }

Parameters

instance_name
specifies a unique name for an instance in the cluster.

status
displays status information for the instance. Values are:

- Up
- Down
- Undefined
- Invalid
- Start
- Init
- Quiesce

log
displays the instance log.

errors [ minseverity severity_level ]
displays log file entries for errors. Optionally, limit displayed error entries to a severity level and above.

Note Error severities_level is an attribute of Adaptive Server error messages, not sybcluster messages.

startdate [ date_string ]
displays log file entries that occur on and after the date specified. The format for date_string is: mm:dd:yy.
If a startdate or enddate date_string is not specified, date_string defaults to the current day.
show instance

denddate [ date_string ]
displays log file entries that occur on or before the date specified. The format
is: mm:dd:yy.

last num_lines
Limits the number of lines displayed, counting backwards from the last line
in the log file.

Examples

Example 1 Displays information about “ase1.”

    show instance ase1 status

    Id   Name    State
    --------    --------
    1       ase1    Down

Example 2 Displays configuration information for “ase1.”

    show instance ase1 config

    Instance: ase1 at blade6:25001

    Private Primary Network
        Address: blade1
        Port Range: 2541 - 2556
        Sybase home: /sybase/sybase_sdc
        ASE home: /sybase/sybase_sdc/ASE-15_0
        Config file: /sybase/sybase_sdc/ase1.cfg

    Private Secondary Network
        Address: blade1
        Port Range: 2557 - 2572

    Log Path: /blade1/sybase/
        ASE-15_0/install/mycluster_ase1.log

Usage

- show instance status displays one of seven different states for the named
  instance:
  - Down
  - Init
  - Invalid
  - Quiesce
  - Start
show membership mode

Description: Displays the cluster’s current membership mode. Membership mode specifies whether or not Veritas Cluster Integration is supported on the current cluster.

Syntax: show membership mode

Usage: Values for show membership mode are:

- vcs – VCS is supported for the current cluster.
- native – VCS is not supported for the current cluster.

- If the cluster is running in vcs membership mode, make sure you shut down or start up servers and the cluster using VCS shutdown and start-up mechanisms.

show monitorserver config

Description: Displays the Monitor Server name, listening port number, node name, and instance name.

Syntax: show monitorserver config

Usage: Use the sybcluster show monitorserver config command to show Monitor Server information.
show session

Description Displays current discovery and agent information.

Syntax show session

Examples Displays agent status information.

show session

Session information

---------------
Sybase sybcluster Command Line Utility/15.0.1/CE GA
2/S/jdk1.4.2/sybclustermain/129/Mon Aug 13 09:59:51 PDT 2007

Connected Cluster: mycluster

Default Cluster:

Default Instance:

Agent Specifications:
[1]: oddjob:7171

Discovery Specifications:

Agent Connections: 1


Node Name: oddjob1
Agent Port: 7171
Agent Version: 2.5.0
Agent Build: 980
OS Name: Linux
OS Version: 2.6.9-42.ELsmp
OS Architecture: amd64

Agent Service Info:

Agent Service (Agent) Build:980 Status:running
BootstrapService (BootstrapService) Build: <unavailable> Status: running
Configuration Service (ConfigService) Build: 980 Status: running
Deployment Service (DeploymentService) Build: 19 Status: running
Environment Service(EnvironmentDiscoveryService) Build: 980 Status: running
File Transfer Service (FileTransferService) Build: 980 Status: running
Plugin Registration Service (PluginRegisterService) Build:980 Status:
CHAPTER 8  Using sybcluster to Manage the Cluster

running
RMI Service (RMIService) Build: 980  Status: running
Remote Shell Service (RemoteShellService) Build: 980  Status: running
Security Service (SecurityService) Build: 980  Status: running
Self Discovery Service (SelfDiscoveryService) Build: 980  Status: running
Service Registration Service (ServiceRegistrationService) Build: 980  Status: running
Session Service (SessionService) Build: 980  Status: running
Sybase Home Service (SybaseHomeService) Build: 14  Status: running
Agent Plugin Info:
ASE Cluster Agent Plugin(com.sybase.ase.cluster)
  Version: 15.0.1 Build: 129 Instance: 1
  Status: running
Cluster Name: mycluster
Env Shell: /oddjob1/work2/
  sybase_sybclustermain_mycluster_vu/SYBASE.sh
  Shell Type: sh
Sybase Home: /oddjob1/
  work2/sybase_sybclustermain_mycluster_vu
ASE Home: /oddjob1/
  work2/sybase_sybclustermain_mycluster_vu/ASE-15_0
ASE Version: Adaptive Server Enterprise/
  15.0.1/EBF 14721 Cluster Edition/B/x86_64/
  Enterprise Linux/asecluster3/2381/64-bit/
  FBO/Mon Nov 12 07:44:23 2007
ASE Login:  sa
Update time: 300 seconds

Usage
Use the sybcluster show session command to view information about the current cluster.
**show xpserver**

**Description**
Displays the XP Server name and listening port number, node name, and instance name configured on each node.

**Syntax**
show xpserver

**Examples**
Displays the XP Server name, listening port number, node name, and instance name.

    show xpserver config

    **XP Server configuration for the cluster**

    | XPServer Name | Port | Host   | Node |
    |---------------|------|--------|------|
    | ase1_XP       | 4010 | blade1 | ase1 |
    | ase2_XP       | 4011 | blade2 | ase2 |
    | ase3_XP       | 4012 | blade3 | ase3 |

**Usage**
Use the sybcluster show xpserver command to view information about XP Server.

**shutdown cluster**

**Description**
Shuts down the cluster by executing a Transact-SQL shutdown command for each instance in the cluster’s instance list, in the order specified in the cluster configuration file.

**Syntax**
shutdown cluster [ nowait ]

**Parameters**
nowait
    shuts down the cluster immediately, without waiting for transactions or statements currently executing to conclude. By default, sybcluster waits for all transactions and statements to execute before shutting down the cluster.

**Examples**
Shuts down the current cluster.

    shutdown cluster

INFO - ...
INFO - 01:00:00000:00117:2007/06/02 00:23:53.56 kernel uesshutdown: exiting
Usage

- sybcluster prompts for confirmation before shutting down the cluster.
- If the cluster is managed by VCS, shutdown cluster fails. You must use VCS shut down mechanisms to shut down the cluster.

shutdown instance

Description
Shuts down the instance by executing a Transact-SQL shutdown command.

Syntax

```
shutdown instance [ instance_name ] [ nowait ]
```

Parameters

- `instance_name` is the unique name of an instance in the cluster.
- `nowait` shuts down the instance immediately, without waiting for currently executing transactions or statements to finish.

Examples

Shuts down the instance “ase1,” after waiting for currently executing transactions or statements to finish.

```
shutdown instance ase1
```

```
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel shutdown server ase1
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 Server SHUTDOWN by request.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 ASE is terminating this process
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 shut down local cluster server.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel coordinator to be shutdown, newcco is 0.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel Single server cluster.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel cipcnode_down(): Node 1 down event.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 server ASE shutdown by request.
INFO - 01:00:00000:00113:2007/06/02 00:31:24/14 kernel ueshutdown: exiting
```

Usage

- Shutting down the last instance in a cluster also shuts down the cluster.
- sybcluster prompts for confirmation before shutting down the instance.
If the cluster is managed by VCS, shutdown instance fails. You must use VCS shut down mechanisms to shut down the instance.

### start cluster

**Description**  
Starts all instances in the cluster.

**Syntax**  
start cluster

**Examples**  
Starts the current cluster.

```plaintext
  start cluster

  INFO - [cluster boot log]
  ...
  INFO - 02:00:00000:00002:2007/06/02 00:21:53.56 server 'ase1' (ID=1).
  INFO - 02:00:00000:00002:2007/06/02 00:21:53.56 server Master device size: 80 megabytes, or 40960 virtual pages.
```

**Usage**  
You must connect to the cluster before starting it.

### start instance

**Description**  
Starts an instance.

**Syntax**  
start instance [ instance_name ] [ unlock ]

**Parameters**  
- `instance_name`  
  specifies a unique name for an instance in the cluster. If you do not enter a cluster name, sybcluster uses the instance specified in the sybcluster command line or specified with the use command.

- `unlock`  
  removes the lock from a cluster that has been terminated unexpectedly. The cluster must be Down before using unlock.

**Warning!** Do not use the unlock parameter unless you have verified that all instances in the cluster are shutdown.
Usage

The instance must be Down to use start instance unlock.

**upgrade server**

**Description**

Upgrades a non-clustered Adaptive Server to the Adaptive Server Cluster Edition, and creates a cluster with a single instance. You can perform the upgrade by answering prompts at the command line or via an input file.

You can upgrade from Adaptive Server version 12.5 and later to the Cluster Edition.

**Syntax**

```
upgrade server server_name
    [ login login_name ]
    [ password password ]
    [ agent agent_spec ]
    [ discovery discovery_spec ]
    [ file input_file_name ]
    [ checkonly ]
```

**Parameters**

- `server_name` is the name of the non-clustered Adaptive Server.

- `login login_name`

  is the management agent login for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.

- `password password`

  is the management agent password for the Sybase Common Security Infrastructure in the Unified Agent framework. The default user name after installation is “uafadmin” with no password; this is the Simple Login Module in the Agent configuration. The user name and password can be configured to use several different mechanisms for authentication and authorization, including operating system logins.
agent \textit{agent\_spec}  

is the agent specification that identifies the node in the cluster running a Unified Agent, and the port number that \texttt{sybcluster} uses to connect the Unified Agent. When upgrading a non-clustered Adaptive Server, there is only one node. The format for \textit{agent\_spec} is \texttt{“node\_name:port\_number”}. The default port number is “9999.”

discovery \textit{discovery\_spec}  

is the discovery method used to identify the agents responsible for the requested cluster. The format is \texttt{“method[(method\_specification)]”}. See Table 8-1 for more information about discovery methods.

file \textit{file\_name}  

is the input file containing values required for upgrading the server.

checkonly

performs a check run of the non-clustered Adaptive Server to determine its readiness for upgrade.

\textbf{Examples}  

\textbf{Example} Upgrades “myserver” to the Cluster Edition:

\begin{verbatim}
upgrade server

Enter the name of the cluster: new_cluster
Enter the existing Sybase installation directory for server myserver:
Enter the name of the subdirectory containing the ASE installation for server myserver:
Enter the name of the subdirectory containing the OCS installation for server myserver:
Enter the name of an sa login on server exit: [sa]
Enter a password:
Cluster new_cluster - Enter the maximum number of instances: [4]
Verifying the supplied agent specifications...
1>tigger 9999 2.5.0 Linux
Enter the number representing the cluster node 1 [1]
Will this cluster be configured using private SYBASE installations? (Y/N)
...
\end{verbatim}

The information required to upgrade a server or create a cluster are the same. See “create cluster” on page 292 or the installation guide for your platform.

\textbf{Usage}  

- \texttt{upgrade server} prompts for these values:
  - The Sybase installation directory of the non-clustered Adaptive Server
  - The release home directory of the non-clustered Adaptive Server
  - The Open Client home directory of the non-clustered Adaptive Server
• The installation mode, private or shared
• If `sybcluster` detects a VCS subsystem, asks if you want to include VCS integration with the cluster
• If you have not configured LDAP, the path to the interfaces file
• The name of the first instance in the cluster
• Other values as required to create a cluster
• Use of the `checkonly` option does not perform any upgrade steps; it does check the server’s readiness for upgrade. Any error conditions found by `checkonly` must be resolved before actually performing the upgrade.

### `use`

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies the default instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td><code>use instance_name</code></td>
</tr>
<tr>
<td>Usage</td>
<td><code>use</code> overrides the instance name specified in the <code>sybcluster</code> command line.</td>
</tr>
</tbody>
</table>
Chapter 9

Migration Utility

This chapter discusses sybmigrate.

### Topic | Page
---|---
Overview | 327
Before you begin | 331
Migration process | 334
Sybmigrate with Encrypted Columns | 359
Post-migration activities | 360
Migrating databases in the Replication Server domain | 361
Limitations | 368
Troubleshooting and error messages | 370

**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.
Overview

- 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
Overview

- 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 168.
Before you begin

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

*syb*mi*grate* works on both UNIX and Windows platforms.

- For UNIX, the executable file is located in
  `$SYBASE/$SYBASE_ASE/bin/sybmi*grate*`.
- For Windows, the executable file is located in
  `%SYBASE%\%SYBASE_ASE%\*in\sybmi*grate*.*at*`.

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 `sybmi*grate*` 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

*syb*mi*grate* 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.
Migration process

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.

sybmiigrate 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:

• sybmiigrate 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 sybmiigrate” on page 341.

• To speed the migration process, you can run multiple sessions of sybmiigrate within the same server. However, running more than one session of sybmiigrate 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 sybmiigrate 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.

• To migrate an Adaptive Server using single-byte character sets to an Adaptive Server using multibyte character sets (utf8):
  a  Use `sybmigrate` to migrate to a server using the same, single-byte character set.
  b  Change the character set to multibyte after migration is complete.

`sybcluster` cannot migrate data directly from an Adaptive Server using single-byte character sets to an Adaptive Server using multibyte character sets.

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 the 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 `sysaltuages` 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:

```
symbmigrate [-v ] [-h ] [-f ]
[-D 1 | 2 | 3 | 4 ]
[-I 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.

- `-I` 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 `-I` 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:
      \texttt{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:
      \texttt{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:
      \texttt{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.
    - 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.
Migration process

- 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.

- **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.

  -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.

* 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 -I command line argument.

If you are not using the interfaces file, you cannot use the -I 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.

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.
Migration process

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.
Migration process

You can 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:

- The `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:

```plaintext
[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:

```plaintext
# This is a sample Migration Tool resource file.
# This resource file will migrate objects in pubs2,
# pubs3, and foo databases.
#
```

```
# Server wide information
```

```plaintext
# This is a sample Migration Tool resource file.
# This resource file will migrate objects in pubs2,
# pubs3, and foo databases.
# Server wide information
```
Migration process

[server]
# "<host name>:<port number>" or just server name.
soure_ase=tho:5002
soure_ase_login=sa
soure_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=5
work_database_device=master

data_copy_thread=5
create_index_thread=1
work_thread=10

# Automatically select the dependent objects for migration and
# validation. Valid values are "yes" or "no".
auto_select_dependent_objects=yes

# 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
Migration process

# 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.aunmind
publishers.pubind
roysched.titleidind
sales.salesind
salesdetail.titleidind
salesdetail.salesdetailind
titleauthor.taind
titleauthor.auidind
titleauthor.titleidind
titles.titleidind
titles.titleind

deltitle
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

data_copy_thread=5

create_index_thread=1

work_thread=10

# Migrate objects
Migration process

user_defined_type_create_list=<ALL_OBJECTS>

default_create_list=<ALL_OBJECTS>

rule_create_list=<ALL_OBJECTS>

table_create_list=
doauthors
publishers
do.titles
stores
do.sales
do.store_employees
salesdetail
do.titleauthor
do.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
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 encrypted columns in ciphertext format by default. This avoids the overhead of decrypting data at the source and reencrypting it at the target. In some cases, however, sybmigrate chooses the reencrypt method of migration, which does decrypt data at the source and reencrypts it at the target.

3. 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.

4. 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.
**Post-migration activities**

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.

❖ **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 362 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 364.

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 364.

❖ 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 setttrunc ("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 364.

**Restoring the RSSD**

This is the procedure for restoring the RSSD.

❖ **Restoring the RSSD**

1 If the RSSD is a primary database, follow the procedure in “Restoring primary databases” on page 362.

   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 Repliction Server. Log in to Replication Server and enter:
   ```
   sysadmin hibernate_off replication_server
   ```
Migrating databases in the Replication Server domain

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.

❖ 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
```
+ "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.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Default</th>
<th>Config Value</th>
<th>Run value</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>fade timeout</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>scan timeout</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>retry timeout</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>rs username</td>
<td>n/a</td>
<td>rsl1_user</td>
<td>rsl1_user</td>
</tr>
<tr>
<td>trace flags</td>
<td>0</td>
<td>8194</td>
<td>8194</td>
</tr>
<tr>
<td>batch ltl</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>rs servername</td>
<td>n/a</td>
<td>rsl1</td>
<td>rsl1</td>
</tr>
<tr>
<td>send buffer size</td>
<td>2k</td>
<td>2k</td>
<td>2k</td>
</tr>
<tr>
<td>trace log file</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>connect database</td>
<td>n/a</td>
<td>pdb1</td>
<td>pdb1</td>
</tr>
<tr>
<td>connect dataserver</td>
<td>n/a</td>
<td>pds1</td>
<td>pds1</td>
</tr>
<tr>
<td>can batch size</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>security mechanism</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>msg integrity</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>unified login</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>kip ltl errors</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>msg origin check</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>short ltl keywords</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>msg confidentiality</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>data limits filter mode</td>
<td>stop</td>
<td>stop</td>
<td>stop</td>
</tr>
</tbody>
</table>
This is a list of explicitly replicated tables. sybmigrate 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Repdef</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>owner</td>
<td>off</td>
</tr>
<tr>
<td>t2</td>
<td>owner</td>
<td>on</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Log</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>function</td>
<td>log_sproc</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>function</td>
<td>log_current</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>table</td>
<td>log_sproc</td>
<td></td>
</tr>
<tr>
<td>p4</td>
<td>table</td>
<td>log_current</td>
<td></td>
</tr>
</tbody>
</table>

(4 rows affected)

This is information about the secondary truncation page. You will need the generation_id column during the post-migration steps.

<table>
<thead>
<tr>
<th>database_id</th>
<th>database_name</th>
<th>dbrepstat</th>
<th>generation_id</th>
<th>ltl_version</th>
</tr>
</thead>
<tbody>
<tr>
<td>621</td>
<td>pdb1</td>
<td>1</td>
<td>167</td>
<td>0</td>
</tr>
</tbody>
</table>

(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 9-1.

<table>
<thead>
<tr>
<th>Replication Server version</th>
<th>LTL version</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1 and earlier</td>
<td>&lt; 400</td>
</tr>
<tr>
<td>12.5 and later</td>
<td>&gt;= 400</td>
</tr>
</tbody>
</table>

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:
Limitations

- **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 9-2 shows column and width limits for Replication Server 12.1 and earlier and Replication Server 12.5 and later.

<table>
<thead>
<tr>
<th>Property</th>
<th>Replication Server 12.1 and earlier</th>
<th>Replication Server 12.5 and later</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column count</td>
<td>250</td>
<td>65535</td>
</tr>
<tr>
<td>Column width</td>
<td>255</td>
<td>65535</td>
</tr>
</tbody>
</table>

**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.

- **Stopping high availability before beginning database migration**
  1. Decouple primary and secondary Adaptive Servers.

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_threasholdaction 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.

❖ 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 inJvmShared 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).
Symbols
!! (exclamation points) operating system commands prefix (isql) 216
 ::= (BNF notation)
    in SQL statements xviii
 , (comma)
    in SQL statements xviii
{ } (curly braces)
    in SQL statements xviii
() (parentheses)
    in SQL statements xviii
[ ] (square brackets)
    in SQL statements xviii
< (redirect in) in isql 15
> (redirect out) in isql 15
\ (backslash)
    escaping special characters 118
\ (backslash) data field terminator in interactive bcp 72
\0 (null) character terminator in interactive bcp 72
" " (enclosing special characters) 118

A
actions
collect 240
compare 240
custom start 240
fix_stats 239
simple start 239
start_stats to fix missing statistics 237

Adaptive Server
configuration of, for migration 338
executing (dataserver) 161
executing (sqlsrvr) 256
rolling back processes 86
upgrading (sqlupgrade) 233, 263
upgrading with resource files (sqlupgraderes) 264
add instance 287
adding
network transport addresses in dsedit 111, 113
server entries in dsccp 96
server entries in dsedit 108, 110
additional network memory parameter 336
allow_dup_row option to create index, and bcp 90
application programs, copying data from 79
archive database access
downgrading 341
migrating 334, 340
ASCII format, bcp and 62
csv (character set)
defncopy and 193

B
backing up, compared to bulk copying 92
backslash (\)
    escaping special characters 118
    terminator in interactive bcp 72
Backup Server
    See also backupserver utility command 124
backup Server 124
backupserver utility command 124
cache size, max for
    defined 124
character set, default 125
defin ed 124
terminal in interactive bcp 72
full path names, specifying 125
interfaces file 128
interfaces file, name and location of 124
language, default 125
LC_ALL environment variable 125
permissions required for 129
server connections, number of 124
server names, specifying 124
starting servers and 128
Index

- trace flags 128
- Backus Naur Form (BNF) notation xvii, xviii
- -bbatch_size parameter 86
- bcp 46
- bcp utility command 92, 130
- allow_dup_row option to create index 90
- ASCII format and 62
- batch operations in 86
- batch size settings 133
- -bbatch_size parameter 86
- binary format and 43
- character format 63
- character format files 62
- character format, default 63
- character formats accepted 43
- character set defaults 136
- configuration parameters 89
- copying data in 52, 55, 82, 85
- copying data out 79, 81, 88
- copying in batches of table rows 86
- data integrity 91
- data loss and dumping 51
- data storage size in 66
- data transfer, preparing for 44
- datatypes 71
- default values for data 91
- defaults for columns and datatypes 91
- defined 130
- described 42
- dump database and 92
- error files 89
- errors in data conversion 89
- examples for using 135
- fast version of 50
- fast version of, and data recoverability 49
- field lengths 62
- field terminators 64, 66
- file storage types in 66
- format files 74, 78
- IDENTITY columns and 60
- improving performance of operations 89
- improving recoverability when copying data in 87
- index creation 90
- indexes and triggers, dropping 52
- insert and 92
- interactive mode 64, 72
- languages, using alternate 85
- load database and 92
- load transaction and 92
- maximum speed, enabling 91
- native file format and 62
- native format option and 62, 63
- non-character to character datatype default field lengths 69
- non-interactive 62
- non-iso_1 data files and 85
- other Adaptive Server utilities and 91
- page allocations, increasing for 55
- partitioned tables and 53, 61, 88
- password encryption in 137
- performance issues with 48, 60, 91
- permissions required for 44, 68
- prefix length 66
- prompts and responses. See Interactive bcp 64
- prompts in 66
- rolling back processes 86
- row terminators 64
- rules and copying data 91
- select into/bulkcopy/pllsort and 52
- slow version and deadlocks on index pages 54
- slow version of 50
- sp_dboption 52
- space needed for data 52
- special characters, handling 134
- speed, and indexes and triggers 49
- speed, modes of 45
- storage types 71
- system data format (SDF) output in 79
- system differences, operating 63
- table defaults and copying data 91
- terminators used in 43
- transferring data between programs 42
- triggers and data copying 91
- triggers not fired on target table 49
- unique IDENTITY column values and 60
- using with row-level access rules 86
- warning about data recovery 133
- binary data
  - bcp and 43
- default in interactive bcp 68
- BNF notation in SQL statements xvii, xviii
- brackets. See square brackets [ ]
buffer, query 12
buildmaster utility command. See dataserver
bulk copy process
    See also bcp utility command 130

C
carriage-return data field terminator (\r) in interactive bcp 72
case sensitivity
    in SQL xix
car datatype, and bcp 62
character formats
    bcp 43
default in bcp 62, 63
terminators for 71
character set
    specification for migration 344
character sets
    backupserver, default in 125
    bcp, defaults in 136
    converting from non-character data 69
    installing and modifying (sqllocres) 255
    installing and modifying, in GUI (sqlloc) 254
    loading, with charset 158
    platform default 85
charset utility command 158
    defined 158
    permissions required for 158
    settings for 158
CIS bulk insert array size
    configuration for migration 335, 339
CIS bulk insert batch size 339
CIS packet size
    configuration for migration 338
    for migration 336
closing a session
    dscp 94
desedit 106
collect action of QPTune 240
column precision in numeric or decimal storage formats 78
column scale in numeric or decimal storage formats 78
columns

datatype sizes and 69
default values and bcp 91
fixed- and variable-length 68
null 89
separator character (isql) 11
comma (,)
in SQL statements xviii
comma-delimited output 79, 81
command terminator (isql) 10
statistics option interaction 14
compare action of QPTune 240
component directory for migration 333
connect 290
conventions
    See also syntax
    Transact-SQL syntax xvii
    used in the Reference Manual xvii
copy threads on sybmigrate 338
copying
    between sessions in dscp 98
different sessions in dscp 98
    new entries with dscp 98
    server entries with dscp 97
    server entries with desedit 110, 111, 112, 113
copying data in
    batch operations in 86
    improving recoverability after rolling back 87
    parallel bcp, requirements for 55, 60
    partitioned tables 53
    partitions, random use of 54
    steps, using fast version of bcp 52
    copying data in with interactive bcp 82, 85
    compatibility of datatypes, and failure 84
delimiters 84
    error files and 90
    field lengths 82
copying data out with interactive bcp 79, 81
delimiters 80
    error files and 90
    fixed-length fields 79
    for other software 79
text and image data 88
copying definitions 192
create cluster 292
create index command, bcp and duplicate rows 90
creating
Index

new servers (srvbuild) 265, 266
new servers using resource files (srvbuildres) 267
curly braces ({} in SQL statements xviii
custom start action of QPTune 240

data
changing with Adaptive Server commands 92
conversion errors 89, 90
importing and exporting with bcp 42
moving with Adaptive Server commands 91
padding with spaces in interactive bcp 69
parsing. See Field terminators 64
permission required to copy into tables 44, 144
recoverability 49
transferring from different programs using bcp 42
data transfer, default formats with bcp 62
database data, migration of 329
database management systems, other 79
database objects
  copying using bcp 130
databases, copying with bcp 92
dataserver utility command 161
    defined 161
    passwords, generating new 166
    permissions required for 166, 260
datatypes
  bcp file storage types for 65, 68
  bcp format files for 74
  bcp, used in 71
  char 62
  copying and compatibility 84
  default values and bcp 91
  field lengths in interactive bcp 66, 72
  implicit conversions of 66, 67
  non-character to character default field lengths in bcp 69
  storage (SYB) 76
debug level for symigrate 342
debugging utility 249
default network packet size configuration parameter 89
defaults
  bcp data conversion 69
  bcp prompts 64, 72
character sets in backupserver 125
character sets, in bcp 136
copying into tables using data 91
copying, with defncopy 192
languages in backupserver 125
prompts in interactive bcp 64, 72
select into/bulkcopy/pllsort option settings in new databases 52
definitions
backupserver utility command 124
bcp utility command 130
charset utility command 158
copying. See defncopy utility command 192
dataserver utility command 161
defncopy utility command 192
dscp utility command 198
dedit utility command 199
isql utility command 207
optdiag utility command 222, 229
showserver utility command 248
sqlloc utility command 254
sqllocres utility command 255
sqlsrvr utility command 256
sqlupgrader utility command 263
sqlupgraderes utility command 264
srvbuild utility command 265
srvbuildres utility command 267
startserver utility command 268
symigrate utility command 271
xpserver utility command 277
defncopy utility command 192
ascii_7 character set and 193
defined 192
encrypted text 197
examples 195
failure due to long comments 196
passwords and crashing 192
Report Workbench, incompatibility with 192
deleting
  server entries (dscp) 99
delimiters
  copying data in with 84
copying data out with 80
deploy plugin 295
diagnose cluster 297
diagnose instance 298

Adaptive Server Enterprise
directory services
  entries, adding 108
  entries, copying to 110, 112, 113
  entries, deleting 109
  entries, modifying 108
  entries, renaming 109
  opening to, with dsedit 105
disconnect 299
DISPLAY environment variable
  setting, in dsedit 104
downgrading an archive database 341
drop cluster 300
drop instance 301
dscp utility command 198
  commands, table of 100
  defined 93, 198
  entries, creating new, by copying 98
  examples of 198
  exit command 100
  help 93, 100
  permissions required for 198
  quit command 100
  server attributes 95
  server entries, adding 96
  server entries, copying 97
  server entries, deleting 99
  server entries, listing 98
  server entries, modifying 96
  server entry contents, viewing 99
  sessions, closing 94
  sessions, copying between 98
  sessions, copying to different 98
  sessions, listing 94
  sessions, opening 94
  starting 93
dsedit utility command 199
  about 103
  command line arguments 103
  defined 199
  Directory Service Session screen 106
  DISPLAY environment variable and 104
  does not start 114
  libtcl.cfg file 105
  network transport addresses, adding 111, 113
  network transport addresses, editing 111
  opening with sql.ini 105
  permissions required for 104, 199
  remote machines, running from 104
  Select a Directory Service screen 104
  server attributes 107
  server entries, adding 108, 110
  server entries, cannot add, modify, or delete 114
  server entries, copying 110, 111, 112, 113
  server entries, deleting 109
  server entries, modifying 109, 110
  server entries, renaming 109
  sessions, closing 106
  sessions, opening 106
  starting 104
  starting from command prompt 103
  starting from Windows Explorer 103
  SYBASE environmental variable and locating
    libtcl.cfg 105
    troubleshooting 103, 114
    xd2 Unable to open X displayxd3 error message
    114
  DSLISTEN environment variable
    backupserver and 128
  dump database command
    bcp and 92
  dump transaction command
    bcp and 92
    error message recommending use of dump database 49
E
  echo input (isql) 15
  editing
    interfaces files 104, 106
    interfaces files in GUI (dsedit) 199
    network transport addresses in dsedit 111
  editing interfaces files in GUI
    See also dsedit utility command 199
  enable unicode conversions for migration 336
  encrypted hidden text 197
  environment variables
    DSLISTEN (backupserver) 128
    LANG (backupserver) 125
    LC_ALL (backupserver) 125
    release path, for sybmigrate 333
host file
  isql and reading  15
host files
  column order    76
data length  78
data type storage formats (SYB)  76
  interactive bcp native format  69

IDENTITY columns
  bcp and  60
  parallel bcp and  60
ignore_dup_key option, create index, and bcp  90
image datatype
  interactive bcp and  68
importing data. See Copying data in with interactive bcp  82
index re-creation for migration  360
index threads for sybmigrate  338
indexes
  bcp, dropping before using  52
  slowing down bcp  49
insert command
  bulk copying, comparing  92
installing
  character sets (sqllocres)  255
  character sets, in GUI (sqlloc)  254
  languages (sqllocres)  255
  languages, in GUI (sqlloc)  254
  languages, new (langinstall)  219, 221
  sort orders (sqllocres)  255
  sort orders, in GUI (sqlloc)  254
intermediate bcp  64, 72
backslash terminator (\)  72
binary data, default  68
carriage-return data field terminator (\r)  72
character format files, terminators for  71
copying data for other software  65
copying data in  82, 85
copying data out  79, 81
defaults for prompts  64, 72
field length  69, 71
field terminators  71, 72
fields, prefix length of  68, 69
file storage types  65, 68
image datatype, default for  68
implicit conversion of datatypes  66, 67
null or invisible character terminator (\0)  72
null values and  68
padding data with spaces  69
row terminators  71, 72
storage length  69, 71
tab data field terminator (\t)  71
terminator for new lines (\n)  72
terminators for tabular data preparation  71
terminators, field and row  71, 72
text data copying, default for  68
interactive SQL parser  207
  See also isql utility command  207
interfaces file
  for sybmigrate  342
interfaces files
  backpuserver and  128
  backpuserver, specifying names and locations of  124
dscp sessions, opening  94
dscp, viewing and editing with  198
dsedit, editing with  106
dsedit, viewing and editing in GUI with  199
dsedit, viewing and editing with  199
  opening for editing  104
iso_1 character set  85
isql utility command  207
  < (redirect in) symbol  15
  << symbol  215
  > (redirect out) symbol  15
column-separator character  11
command terminator  10
command terminator, changing  13
command terminator, resetting  209
  correcting typing errors  12
defined  207
echo input  15
examples of  208
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

max network packet size configuration parameter 89
max network packet size, configuration for migration 339
max packet size allowed, for sybmigrate 336
max parallel degree, configuration for migration 339
merging databases, during migration 371
migrate session, for sybmigrate 334
migrating
servers (sybmigrate) 271
migrating an archive database 334, 340
missing statistics
fix_stats action 239
start_stats action 237
modifying
character sets (sqllocres) 255
character sets, in GUI (sqlloc) 254
languages (sqllocres) 255
languages, in GUI (sqlloc) 254
server entries (dsoc) 96
server entries with dsedit 109, 110
sort orders (sqllocres) 255
sort orders, in GUI (sqlloc) 254

native file format
bcp and 62
native format files 62
network packet size
isql, specifying in 15
network transport addresses
adding with dsedit 111, 113
dsedit, editing with 111
network transports 106
new servers, creating 265, 266
See also srvbuild utility command 265
new servers, creating, using resource files 267
See also srvbuildres utility command 267
new-line terminator (\n)
in bcp 63
in interactive bcp 72
noncharacter datatypes, operating system format for 68
non-printable characters, host file 69
notes
ascii_7 character set compatibility 193
batch size settings in bcp 133
defncopy and Report Workbench 192
hidden encrypted text 197
null terminator versus no terminator 72
passwords, using 118
security 118
select into/bulkcopy/pllsort and copying out data in bcp 52
slow version of bcp and deadlocks 54
special characters, handling in bcp 134
system differences in bcp, operating 63
triggers not fired by bcp in target table 49
null character terminator (\0) in interactive bcp 72
null values 89
interactive bcp and 68
number (quantity of)
system connections to backupserver 124
number of sort buffers for migration 339
number of user connections for migration 338, 339
number of worker processes for migration 339
numbers line, removing from isql 11, 15
numeric datatypes
bcp conversion to character storage 69
column precision 78
column scale 78
operating system format for 63

O
object hierarchy for migration 334
objects failed migration of 370
re-creating after migration 372
opening sessions in dscp 94
sessions in dsedit 106
operating system files
native format 62
permission required for copying tables into 44, 144
operating systems commands prefix (!) (isql) 216
non-character datatype formatting 68
numeric datatype formatting 63
optdiag command, changes 227
optdiag utility command 222, 229
defined 222, 229
permissions required for 228
output formats, data. See Copying data out with interactive bcp 79

P
packet size, network
bcp, specifying with 88
isql, specifying in 15
specifying in isql 15
padding data and bcp 71
parallel bcp
  copying to a specific partition 55, 59
different methods of using 58, 60
IDENTITY columns and 59
syntax 59
parameter
  --hide-vcc 143
parentheses ()
in SQL statements xviii
partitioned tables in bcp 53, 61
copying data into randomly 54
copying data into, methods for 53
page allocations, increasing 55
passwords
bcp encryption 137
defncopy and 192
new, generating 166
notes about using 118
performance
bcp issues and 48, 60, 91
bulk copy and packet size 89
isql network packet size and 15
permissions required for backupserver utility command 129
bcp utility command 44, 68
charset utility command 158
dataserver utility command 166, 260
dscp utility command 198
dsedit utility command 199
dsedit, required for 104
Index

langinstall utility command 219
operating system file, copying tables into 44, 144
optdiag utility command 228
sqlloc utility command 254
sqllocres utility command 255
sqlupgrade utility command 263
sqlupgraderes utility command 264
srvbuild utility command 266
srvbuildres utility command 267
tables, copying data into 44, 144
xserver utility command 278
post-migration activities, for sybmigrate 360
prefix field lengths in interactive bcp 68, 69
prefix length 78
prefix lengths in bcp 66
preupgrade utility command 229
primary database, restoring after migration 362
prompts
bcp. See Interactive bcp 64
prompts in bcp utility command 66

R
redirect in symbol (<) in isql 15
redirect out symbol (>) in isql 15
release path, for sybmigrate 333
remigrating a database 372
remote machines 104
remote machines, using dsedit utility on 104
renaming server entries with dsedit 109
replicate database
restoring after migration 363
replicated databases
post-migration procedures 362
Replication Server, migrating replicated data to Adaptive Server 361
report Workbench, incompatibility with defncopy 192
reports for sybmigrate 272, 343
reset command (isql) 12
resource file mode, for migration 352
roll-back processes 86
row terminators
bcp, in 64
interactive bcp, in 71, 72
row-level access rules and bcp 86
rows in bcp, and erroneous table 90
rows, table
bulk copying and failed length 87
rules
defncopy, copying with 192
tables, copying data into 91

S
saving
errors during copy in operations 89
ersors during copy out operations 90
format files (bcp) 74
security Mechanism server attribute (dscp) 96
security, notes about 118
select a Directory Service screen 104
select into command, compared to bulk copying 92
select into/bulkcopy/pllsort database option
bcp and 52
server column
names 78

384

Adaptive Server Enterprise
orders 78
server connections
  backupserver, number of 124
server data, migration of 328
server entries
  adding, with dsedit 108
  contents, viewing with dscp 99
  copying, with dsedit 110, 111, 112, 113
  deleting, with dsedit 109
  dscp, listing with 98
  modifying, with dsedit 109, 110
  renaming, with dsedit 109
  server entries, adding 110
server Entry Editor window 106
server Name server attribute (dscp) 95
server Object Version server attribute (dscp) 95
server Service server attribute (dscp) 95
server Status server attribute (dscp) 95
servers
  backupserver, specifying names in 124
  backupserver, starting in 128
  migrating 271
  modes of speed when using bcp 45
  startserver utility command 268
  startserver, starting in 268
  sybmigrate utility command 271
sessions
  listing (dscp) 94
set cluster 306
set instance 307
setup session, for sybmigrate 334
sever attributes (dscp) 95
show agents 309
show cluster 312
show instance 315
show servers 248
  See also showserver utility command 248
show session 318
showserver utility command 248
  defined 248
  examples of 248
shutdown cluster 320
shutdown instance 321
simple start action of QPTune 239
size
  data storage in bcp 66
  packet size 88
  text or image data 88
slow version of bcp 50
  deadlocks on index pages 54
sort orders
  installing and modifying (sqllocres) 255
  installing and modifying, in GUI (sqlloc) 254
sp_dboption system procedure and bcp 52
space requirements and bcp steps 52
special characters
  bcp, handling in 134
  utility commands, use in 118
specifying full path names in Backup Server 125
specifying server names in backupserver 124
SPX/IPX addresses for interfaces entries 112
SQL parser utility. See isql utility command 207
sql.ini file
  dsedit sessions, opening with 105
  entries, adding 108
  entries, copying to 110, 112, 113
  entries, deleting 109
  entries, modifying 108
  entries, renaming 109
sqlDBG utility 249
sqlloc utility command 254
  defined 254
  permissions required for 254
sqllocres utility command 255
  defined 255
  permissions required for 255
sqlsrvr utility command 256
  defined 256
sqlupgrade utility command 233, 263
  defined 263
  permissions required for 263
sqlupgraderes utility command 264
  defined 264
  permissions required for 264
square brackets [ ]
in SQL statements xviii
svrbuild utility command 265, 266
  defined 265
  permissions required for 266
svrbuildres utility command 267
  defined 267
  permissions required for 267
Index

start cluster 322
start instance 322
starting
   dscp utility 93
desedit utility command 103, 104
servers (startserver) 268
   XP Server manually (xpserver) 277, 278
startserver utility command 268
   backupserver and 128
defined 268
   examples of 268
   runserver file examples 269
statistics
   displaying (optdiag) 222, 229
   isql 14
   loading updated (optdiag) 222, 229
storage types used in bcp 71
stored procedures
   copying with defncopy 192
SYBASE environment variable 333
SYBASE environment variable in dsedit 105
SYBASE_ASE environment variable 333
SYBASE_JRE environment variable 333
sybcluster commands
   add backupserver 287
   add instance 287
   connect 290
   create backupserver 291
   create cluster 292
   create monitorserver 294
   create xpserver 295
deploy plugin 295
diagnose cluster 297
diagnose instance 298
disconnect 299
drop backupserver 299
drop cluster 300
drop instance 301
drop monitorserver 302
drop xpserver 302
exit 303
help 303
localize 303
quit 305
set backupserver 305
set cluster 306
set instance 307
set monitorserver 308
set xpserver port 309
show agents 309
show backupserver config 311
show cluster 312
show instance 315
show monitorserver config 317
show membership mode 317
show session 318
show xpserver 320
shutdown cluster 320
shutdown instance 321
start cluster 322
start instance 322
upgrade server 323
use 325
sybcluster utility 279
sybmigrate
   additional network memory parameter 336
   character set conversion 336
   character set specification 344
   CIS bulk insert array size 339
   CIS bulk insert array size configuration 335
   CIS bulk insert batch size configuration 339
   CIS packet size configuration 336, 338
   command line trace flags 344
   component directory 333
   components for 331
   configuration and tuning 337
   copy threads 338
   debug level 342
   dependencies for 331
   environment variables 333
   errors to avoid 340
   executable file 333
   GUI mode 345
   help information 342
   index re-creation 360
   index threads 338
   installation 332
   interfaces file 342
   Java columns 337
   Java runtime environment 333
   language specification 344
   languages 337

386

Adaptive Server Enterprise
Index

lock session override 342
max packet size allowed configuration 336
max parallel degree configuration 339
memory for the JVM 333
merging two database 371
migrate session 334, 350
migrating Replication Server data 361
migration failure cleanup 372
migration of database data 329
migration of server data 328
migration with high availability 368
multiple sessions 335
number of sort buffers configuration 339
number of user connections configuration 338, 339
number of worker processes configuration 339
object hierarchy 334
objects fail to migrate 370
output template resource file 344
permissions for 332
platforms 333
post-migration activities 360
post-migration procedures for replicated databases 362
pre-migration considerations 335
re-creating an object 372
release path 333
remigrating a database 372
reports 272, 343
resource file mode 352
restoring replicate databases 363
restoring the primary database 362
setup session 334, 345
source Adaptive Server configuration 338
starting 341
sybmigrate log 365
syntax 342
trouble shooting and error messages 370
user-defined log file 344
validate session 334, 351
version string 342
sybmigrate .max network packet size configuration 339
sybmsigrate utility command 271
defined 271
sybmigrate .max memory configuration 336
sybmigrate, what is not migrated 330
SYBMIGRATE_MEMORY environment variable for migration 333
sybmultbuf, using to start Backup Server 125
symbols
in SQL statements xvii, xviii
symbols, field terminator (bcp) 72
syntax
for sybmigrate 342
syntax conventions, Transact-SQL xvii
system data format (SDF) output in bcp 79
system procedure (sp_dboption) 52
T
tab data field terminator (\) in interactive bcp 71
table rows
- copying in batches 86
- length 58
- tables
  - bcp character set defaults 136
  - bcp prompts in 66
  - permission required for copying data into 44, 144
  - permission required to copy into operating system files 44, 144
  - utility commands 118
  - tabular data, copying 72
  - tabular output 79, 81
TCP/IP addresses for interfaces entries 111
terminators (bcp) 78
changing 64
defined 43
field and row, and 71, 72
other programs, using for 71
text datatype
- interactive bcp and 68
- trace flags and backserver 128
- trace flags, for sybmigrate 344

Transact-SQL
- using with isql 10
- transferring data from programs using bcp 42
- transport Address server attribute (dscp) 95
- transport Type server attribute (dscp) 95
- triggers
  - bcp, dropping before using 52
Index

copying with defncopy 192
slowing down bcp 49
tables, copying data into 91
trouble shooting, for migration 370
troubleshooting dsedit 114

U
unlogged transactions 49
upgrading Adaptive Server 233, 263
See also sqlupgrade utility command 263
upgrading Adaptive Server using resource files 264
See also sqlupgraderes utility command 264
use 325
utilities
bulk copy (bcp) 46
utilities, other Adaptive Server, and bcp 91
utility commands
special characters, using 118
table of commands 118

V
validate session for sybmigrate 334
viewing interfaces files in GUI 199
See also dsedit utility command 199
views, copying, with defncopy 192

W
warnings
bcp data loss 51
bcp in data recovery 133
defncopy failure due to long comments 196

X
xd2 xd3 (enclosing special characters) 118
XP Server, starting manually 277, 278
See also xpserver utility command 277
xpserver utility command 277, 278
defined 277