SYBASE®

New Feature Guide

Adaptive Server® Enterprise

12.5.4

DOCUMENT ID: DC00212-01-1254-02

LAST REVISED: July 2006

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

Contents

About This Book

Audience

This manual is for Sybase® System Administrators and Database Owners who are using Adaptive Server® version 12.5.4. It discusses the new features included in this version of Adaptive Server.

How to use this book

This manual includes the following:

- Chapter 1, "Overview" contains overview and summary information about features available in Adaptive Server version 12.5.4.
- Chapter 2, "Security Enhancements" contains information about security enhancements in Adaptive Server version 12.5.4. These enhancements include LDAP, Kerberos, and SSL changes, as well as password complexity enhancements.
- Chapter 3, "Archive Database Access" contains information about the new archive database access feature. Archive database access provides a database administrator with the capability to validate or selectively recover data from a database dump.
- Chapter 4, "Shared Directory Changes" contains information about directory structure changes.
- Chapter 5, "Sybase Driver Support" contains information about the new native driver support.
- Chapter 6, "Dynamically Loading TIBCO Libraries" contains new information about supported TIBCO libraries.
- Chapter 7, "JRE Support" contains information about support for the most recent version of the JRE.
- Chapter 8, "Adaptive Server Plug-in Changes" contains information about changes made to the Adaptive Server plug-in to support archive database access and password complexity.
- Chapter 9, "Changes to Monitoring Tables" contains information about changes to monitoring tables.
- Chapter 10, "Shared Memory in Windows Terminal Server Environments" contains information about using shared memory in a Windows terminal server environment.

- Chapter 12, “Changes to Stored Procedures, Functions, and Commands” includes additional information about changes implemented in Adaptive Server version 12.5.4.

Related documents

The Sybase Adaptive Server Enterprise documentation set comprises these documents:

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

  A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.
- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- New feature documentation for each of the previous 12.5.x releases:
  - *What’s New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.1, the system changes added to support those features, and the changes that may affect your existing applications.
  - *New Functionality in Adaptive Server 12.5.2* – describes the new features in Adaptive Server version 12.5.2, and the system changes added to support those features.
  - *New Feature Guide for Adaptive Server 12.5.3* – describes the new features in Adaptive Server version 12.5.3, and the system changes added to support those features.
  - *New Feature Bulletin for Adaptive Server 12.5.3a* – describes the new features in Adaptive Server version 12.5.3a, and the system changes added to support those features.
- *ASE Replicator User’s Guide* – describes how to use the ASE Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User’s Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.

- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.
- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
  - *Basics* – includes the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how various locking schemas can be used for improving performance in Adaptive Server.
  - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.

    - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL® information:
    - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
    - *Commands* – Transact-SQL commands.
    - *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.
    - *Tables* – Transact-SQL system tables and dbcc tables.
- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Transact-SQL User's Guide* – documents Transact-SQL, the Sybase-enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
- *Using Sybase Failover in High Availability Systems* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.

- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
- *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

Use the Sybase Getting Started CD, the Sybase Technical Library CD, and the Technical Library 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 Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

  Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.
- The Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Updates, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click Certification Report.
3. In the Certification Report filter select a product, platform, and timeframe and then click Go.

4 Click a Certification Report title to display the report.

❖ Finding the latest information on component certifications

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

2 Either select the product family and product under Search by 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.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.

3 Select a product.

4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

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

5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

If you need help

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

CHAPTER 1

Overview

This chapter includes a feature and platform compatibility matrix, as well as summary information about the new features introduced in Adaptive Server versions 12.5.4, 12.5.3a, 12.5.3, and 12.5.2.

Feature and platform compatibility information

Table 1-1 shows feature availability for supported operating systems in Adaptive Server version 12.5.4.

Table 1-1: Adaptive Server features for supported operating systems

Adaptive Server options	Solaris 32-bit	Solaris 64-bit	Solaris x86	Sol Opteron 64-bit	HP-UX PA Risc 64-bit	HP-UX PA Risc 32-bit	HP Tru64	HP-UX Itanium 64-bit	IBM AIX 32-bit	IBM AIX 64-bit	Windows x86	Macintosh OS X	SGI 32-bit	SGI 64-bit	Linux on pSeries	Linux Opteron 64-bit	Linux Itanium 64-bit	Linux x86 32-bit
Encrypted columns	*f*	*f*	–	–	*f*	*f*	–	–	–	*f*	*f*	–	–	–	–	–	–	*f*
High availability	Y	Y	–	–	Y	Y	Y	Y	Y	Y	Y	–	Y	Y	–	–	–	Y
Distributed Transaction Management	Y	Y	–	Y	Y	Y	–	–	Y	Y	Y	–	Y	Y	–	Y	–	Y
XML management	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	–	Y	Y
Java option	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	Y	Y	Y
Native XML	Y	Y	–	Y	Y	Y	–	§	–	Y	Y	–	–	–	–	§	–	Y
Java-based XML	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	–	Y	Y
Web Services	Y	Y	–	–	Y	Y	–	Y	Y	Y	Y	–	Y	Y	–	–	Y	Y

Legend: Y: supported in Adaptive Server 12.5.x; *ƒ*:Introduced in 12.5.3a; ž:Introduced in12.5.4; §: Introduced in 12.5.3 ESD5; Þ: port merge; **–: Not supported**

Adaptive Server options	**Solaris 32-bit**	**Solaris 64-bit**	**Solaris x86**	**Sol Opteron 64-bit**	**HP-UX PA Risc 64-bit**	**HP-UX PA Risc 32-bit**	**HP Tru64**	**HP-UX Itanium 64-bit**	**IBM AIX 32-bit**	**IBM AIX 64-bit**	**Windows x86**	**Macintosh OS X**	**SGI 32-bit**	**SGI 64-bit**	**Linux on pSeries**	**Linux Opteron 64-bit**	**Linux Itanium 64-bit**	**Linux x86 32-bit**
Security and directory services	Y	Y	–	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	Y	Y	Y
LDAP server directory	Y	Y	–	Y	Y	Y	Y	Y	Y	*f*	Y	–	Y	–	–	–	ž	Y
LDAP user authentication	Y	Y	–	–	Y	Y	Y	–	Y	*f*	Y	–	–	–	–	–	–	Y
Secure Sockets Layer	Y	Y	–	Y	Y	Y	Y	ž	Y	Y	Y	–	–	–	–	–	–	Y
Cybersafe Kerberos	Y	Y	–	–	–	–	–	–	Y	–	Y	–	–	–	–	–	–	–
MIT Kerberos	Y	Y	–	Y	*f*	–	–	–	*f*	*f*	–	–	–	–	–	Y	–	Y
Platform Native Kerberos	Y	Y	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Fine grained access control	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	Y	Y	Y
Pluggable Authentication Module	Y	Y	–	Y	–	–	–	ž	–	*f*	–	Y	–	–	–	Y	Y	Y
Content management (eXternal file support)	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	–	Y	Y	Y
Monitor Client GUI	–	–	–	–	–	–	–	–	–	–	Y	–	–	–	–	–	–	–
Enhanced Full-Text Search (EFTS)	Y	Y	–	–	Y	Y	Y	–	Y	Y	Y	–	–	–	–	–	–	Y
Real-time messaging	Y	Y	–	–	–	Y	–	–	–	Y	Y	–	–	–	–	–	–	Y
JMS support	Y	Y	–	–	–	Y	–	–	Y	Y	Y	–	–	–	–	–	–	Y
WebSphere MQ support	*f*	*f*	–	–	–	*f*	–	–	–	*f*	Y	–	–	–	–	–	–	*f*
Disaster recovery	Y	–	–	–	Y	–	Y	Y	Y	–	Y	–	Y	–	–	–	–	Y
Features included in base Adaptive Server																		
Password complexity	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž
Archive database access	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž	ž
IPv6	Y	Y	–	–	*f*	*f*	–	–	–	–	*f*	–	–	–	–	–	–	–
XA libraries	–	–	–	Y	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Cross-platform dump and load	Y	Y	–	Y	Y	Y	–	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Job Scheduler	Y	Y	–	Y	Y	Y	–	–	Y	Y	Y	Y	Y	Y	–	–	–	Y
ASE Replicator	Y	Y	–	–	Y	Y	–	–	Y	Y	Y	Y	–	–	–	–	–	Y

Legend: Y: supported in Adaptive Server 12.5.x; *f*:Introduced in 12.5.3a; ž:Introduced in12.5.4; §: Introduced in 12.5.3 ESD5; Þ: port merge; **–: Not supported**

New feature overview

Table 1-2 includes information about the major features included in Adaptive Server versions 12.5.2, 12.5.3, 12.5.3a, and 12.5.4. For detailed descriptions of these features, see the new feature documentation for the appropriate release.

Table 1-2: New features for Adaptive Server Enterprise version comparison

Adaptive Server feature	Description
Adaptive Server version 12.5.4	
Kerberos enhancements	Adaptive Server includes the following Kerberos enhancements: • New option to specify a Kerberos principal name different from the Adaptive Server name. • Expanded platform support for Kerberos. • You can now use the sp_modifylogin option authenticate with to require Kerberos authentication for an individual login. • sybmapname is a customizable utility to convert external user principal names to the name space of Adaptive Server logins.
LDAP user authentication enhancements	Adaptive Server includes the following LDAP user authentication enhancements: • Automatic failover from authentication against a primary LDAP server to a secondary LDAP server. • Adaptive Server recovers from errors encountered when communicating with the LDAP server. • Enhancements to the communication of password-expiration-related messages obtained from the LDAP server to Adaptive Server clients.
Password complexity enhancements	You can now specify a wide range of password complexity attributes. You can also write your own stored procedures to create site-specific password complexity checks.
Archive database access support	Archive database access provides a database administrator with the capability to validate or selectively recover data from a database dump by making the dump appear as if it were a traditional read-only database.
Shared directory changes	Adaptive Server version 12.5.4 includes a number of changes to the shared directory structure.
Sybase driver support	Adaptive Server now includes the following Adaptive Server Data Providers: • ASE ADO.NET Data Provider 1.1 • ASE OLE DB Provider by Sybase 12.5.1 • ASE ODBC Driver by Sybase 12.5.1

Adaptive Server feature	Description
Dynamically loading TIBCO libraries	TIBCO JMS libraries are now dynamically loaded rather than statically linked to the Adaptive Server executable.
JRE support	Adaptive Server version 12.5.4 includes JRE 1.4.
Adaptive Server plug-in changes	Adaptive Server plug-in supports archive database access and password complexity.
Updating system catalogs	The server-wide command allow updates to system catalogs takes precedence over the stored procedure settings for allow updates.
Monitoring tables changes	Monitoring tables monSysStatement andmonProcessStatement have been changed.
syscomments changes	When the source text of a stored procedure or trigger is stored in the system table syscomments, a query using select * is stored in syscomments expanding the column-list referenced in the select *.
Shared memory support in Windows terminal server environments	To accommodate Windows terminal server shared memory requirements, Adaptive Server version 12.5.4 introduces the new environment variable SYBASE_TS_MODE.
Global login trigger	Adaptive Server version 12.5.4 provides the ability to set a new global login trigger that is executed at each user login.
Exporting set options from a login trigger	Adaptive Server version 12.5.4 enables options set inside login triggers to remain valid for the entire user session.
Adaptive Server version 12.5.3a	
Encrypted columns support	Data encryption in Adaptive Server allows you to encrypt data at the column level.The authentication and access control mechanisms ensure that only properly identified and authorized users can access data. The encrypted columns feature in Adaptive Server is easier to use than encryption in the middle tier or in the client application. No application changes are required when enabling encrypted columns support.
Adaptive Server version 12.5.3	
Secure Sockets Layer enhancements	New SSL CipherSuites supporting Advanced Encryption Standard (defined in FIPS-197) for encrypting network communications. Advanced Encryption Standards allow you to choose whether to use 128- or 256-bit encryption keys.
Support for cross-platform dump and load for databases	Adaptive Server supports databases dumps and loads across platforms with different endian architecture. You can perform dump database and load database from either a big-endian platform to a little-endian platform, or from a little-endian platform to a big-endian platform.
Importing statistics for proxy tables	If the relevant tables and index statistics are available when you perform update statistics on a remote server proxy table, the table catalogs are imported to the local systabstats and sysstatistics.

Adaptive Server feature	Description
top n functionality support	Use the top n clause to limit the number of rows in the result set to the number of rows specified by the integer. Adaptive Server Enterprise supports the top n clause in outer query select statements, but not in the select list of a subquery.
Historical Server enhancements	Adaptive Server allows you to send monitoring data from Historical Server to a database on a specific Adaptive Server.
Real-time data services enhancements	Use sp_configure to set the number for native threads and the wait time for messaging.
Resource governor	Adaptive Server resource limits to help System Administrators prevent queries and transactions from monopolizing server resources. In Adaptive Server version 12.5.3, when System Administrators modify a resource limit, all users logged in the session see the change.
Page allocation for partitioned DOL tables	Adaptive Server version 12.5.3 avoids using extra space by filling up existing allocated extents in the target allocation page even when these extents are assigned to other partitions.
User connections enhancements	User connection enhancements include an updated error message and a correction on reserved sockets.
dtdValidation	The dtdValidate option include three additional parameters: no, yes, and strict.
sybmigrate enhancements	The sybmigrate tool allows you to migrate across versions of Adaptive Server Enterprise and supports source servers from 12.0 through 12.5.3.
New Enhanced Full-Text Search Specialty Data Store language support	EFTS now includes language support for: • Traditional Chinese on the Windows and Solaris platforms • Arabic, Hebrew, Thai, and Russian on the Linux platform
Monitor counters and sp_sysmon	Adaptive Server version 12.5.3 introduces a number of enhancements to improve some of the most commonly used monitoring tools.
Adaptive Server version 12.5.2	
Pluggable Authentication Module support	Pluggable Authentication Modules (PAM) support allows multiple authentication service modules to be stacked and made available without modifying the applications that require the authentication.
Kerberos native library support	Adaptive Server provides the following Kerberos native library support: • Cybersafe Library support • MIT Kerberos library support • Active directory interoperability for user authentication • Kerberos support in jConnect

Adaptive Server feature	Description
Support for password-protected backups	You can protect your database dump from unauthorized loads with the password parameter of the dump database command. If you include the password parameter when you make a database dump, you must also include this password when you load the database.
Access control enhancements	Adaptive Server introduces the following access control enhancements: • Restricted permissions on system catalogs • Improved granularity for set proxy • Grant and revoke for administration commands.
FIPS-140-certified algorithms support	SSL is the standard for securing the transmission of sensitive information, such as credit card numbers, stock trades, and banking transactions over the Internet. It relies on public-key cryptography.
Statement cache enhancements	The statement cache is used for saving SQL cached-statements.
XML services support	The changes in XML services enhance four main areas: • Extended XML query language for the xmlextract built-in function and xmltest predicate. • Enhanced datatype support in the for xml clause of select statements. • Enhanced datatype support for xmlextract, xmlparse, and xmlrepresentation. • Enhanced Java-XML sample code.
Real-time messaging services	Adaptive Server 12.5.2 includes messaging functionality with the Real Time Data Services (RTDS) option package. This option simplifies the development of application that interact with messaging systems and databases.
Web services support	A Web service is a self-contained, modular application that can be accessed through a network connection. Using a Web service, you trade performance for increased interoperability.
IPv6 support	Adaptive Server supports Internet Protocol version 6 (IPv6.)
Enhanced Full-Text Search (EFTS) enhancements	EFTS introduces the following enhancements: • Installation and directory changes • Permission for shutdown • index_any clauses support up to 16000 bytes • New pseudo-column total_docs • Primary keys
Compressed database dump support	The Adaptive Server version 12.5.2 version of dump includes a compression parameter that allows you to compress your database dumps.

Adaptive Server feature	Description
Password-protecting database dumps and loads	The Adaptive Server 12.5.2 version of dump and load database include a password parameter that allows you to password-protect your database dumps.
Failover on Veritas 2.1 support	You can now configuring Adaptive Server on Linux platforms for Failover on Veritas 2.1.
Large memory support	Adaptive Server large memory support on 32-bit Enterprise Linux operating systems increases the amount of available memory in Adaptive Server from 2.7GB to as much as 64GB. Increasing the amount of memory available to Adaptive Server improves performance by significantly reducing the number of times the server must access the disk.

CHAPTER 2 Security Enhancements

Kerberos changes

Kerberos has been enhanced in the following ways:

- You can now specify a server principal name for Kerberos authentication
- The sp_modifylogin and sp_addlogin authenticate with options support Kerberos authentication
- Support for sybmapname
- MIT Kerberos client library support
- Expanded platform support for Kerberos

Specifying the Adaptive Server principal name for Kerberos authentication

The principal name is the name the server uses to authenticate with the Kerberos Key Distribution Center (KDC). When you have multiple instances of Adaptive Server running, you must have different principal names for each Adaptive Server.

In Adaptive Server version 12.5.4, you can use a new data server option and a new environment variable to specify a principal name different from the Adaptive Server name. Adaptive Server name is specified by environment variables DSLISTEN and DSQUERY, or the dataserver command line option `"-s servername"`.

Using SYBASE_PRINCIPAL to specify the Adaptive Server principal name

By default, the principal name is the name of Adaptive Server. To specify a different name, set SYBASE_PRINCIPAL before starting Adaptive Server to use Kerberos:

```
setenv SYBASE_PRINCIPAL <name of principal>
```

Once you have set an Adaptive Server principal name, Adaptive Server uses the value of this variable to authenticate itself with Kerberos.

Using the *-k* data server option to specify the Adaptive Server principal name

You can use the following command line option to specify an Adaptive Server principal name, when starting Adaptive Server.

```
   -k <server principal name>
```

Note If `-k` specifies the Adaptive Server principal name, Adaptive Server does not look at the environment variable.

Using *-k* and SYBASE_PRINCIPAL to set the Adaptive Server principal name

When Adaptive Server is started with the Kerberos security mechanism enabled, Adaptive Server first uses the principal name specified with the `-k` option for Kerberos authentication. If the `-k` option is not specified, Adaptive Server looks for the principal name in the environment variable SYBASE_PRINCIPAL. If neither is specified, Adaptive Server uses the server name for authentication.

In the following example, let the Adaptive Server name be `"ase1254"` and the current realm name be `"MYREALM.COM"`. The Adaptive Server name is specified on the command line with -s parameter to data server. The current realm is specified in *libtcl.cfg* by a `secbase` attribute value:

```
[SECURITY]
csfkrb5=libskrb.so libgss=/krb5/lib/libgss.so
```

```
secbase=@MYREALM.COM
```

The default Adaptive Server principal name is "ase1254@MYREALM.COM". If the principal name defined in the Adaptive Server *keytab* file is "aseprincipal@MYREALM.COM", you can override the default Adaptive Server principal name by setting a server principal name using options 1 or 2 below:

Option 1: ‘-k’ is specified %

```
$SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat
-s ase1254 -k aseprincipal@MYREALM.COM
```

The Adaptive Server principal name used to authenticate with Kerberos is aseprincipal@MYREALM.COM.

Option 2: ‘-k’ is not specified but SYBASE_PRINCIPAL is set

```
setenv SYBASE_PRINCIPAL aseprincipal@MYREALM.COM
$SYBASE/$SYBASE_ASE/bin/dataserver –dmaster.dat
-s ase1254
```

The Adaptive Server principal name used to authenticate with Kerberos is the value of *$SYBASE_PRINCIPAL*, "aseprincipal@MYREALM.COM".

Option 3: Neither ‘-k’ nor SYBASE_PRINCIPAL is set %

```
$SYBASE/$SYBASE_ASE/bin/dataserver –dmaster.dat
-s ase1254
```

The Adaptive Server principal name used to authenticate with Kerberos is ase1254@MYREALM.COM.

For more information about Kerberos, see the Security section of the *System Administration Guide, Volume One*.

sp_modifylogin and *sp_addlogin authenticate with* option

In Adaptive Server version 12.5.4, the authenticate with option to sp_modifylogin or sp_addlogin requires that the login use *only* a specified authentication mechanism. The supported authentication mechanisms are:

- ASE
- LDAP
- PAM
- KERBEROS
- ANY

Using authenticate with with these supported authentication mechanisms allows you to override the server-wide configuration parameters unified login required, enable ldap user auth, and enable pam user auth.

If more than one external authentication mechanism is configured and a login specific authenticate with option has not been set, the external authentication mechanism is decided based on the following order:

1. Kerberos
2. LDAP
3. PAM

For example, if both PAM and LDAP are configured, then LDAP is chosen for external authentication, not PAM.

At the time of authentication, exactly one external authentication mechanism is attempted. Depending on the value of configuration parameters, ASE authentication may be tried when the external authentication fails.

When none of the external authentication mechanisms are configured, Adaptive Server uses ASE authentication.

Using authenticate with to override server-wide authentication options

Note You must have sso_role permissions to modify the authenticate with option for a login.

To configure external authentication mechanisms such as Kerberos, LDAP, or PAM, Sybase recommends that you determine the server-wide authentication setting that matches your company's security policy. This server-wide setting is appropriate for most client connections. Then, you can set individual logins to another authentication mechanism using the authenticate with option.

You can use authenticate with to specify an the external authentication mechanisms Kerberos, LDAP, and PAM. You can also issue sp_modifylogin or sp_addlogin authenticate with to set the authentication mechanism to ASE to use only the Adaptive Server internal authentication mechanism. To allow any authentication mechanism, use authenticate with ANY.

Note When authentication mechanism ANY is set for a login, the login uses the server-wide configuration settings to control authentication. The default authentication mechanism setting for a login is ANY.

sp_modifylogin also checks for any conflicts with any login mapping specified by a previous sp_maplogin. See "Adding tighter controls on login mapping" on page 32 for more details.

Example 1: Creating a local account to run a batch application Consider an environment that uses Kerberos for a centralized user account repository and requires its general population of users to authenticate using Kerberos, Adaptive Server should configure Kerberos by setting parameters:

```
sp_configure  "use security services", 1
go

sp_configure "unified login required", 1
go
```

These configuration parameters now require all user logins, other than login "sa", to authenticate using Kerberos to gain access to the Adaptive Server.

Now consider a nightly batch operation run by the Adaptive Server database administrator or operator, which may authenticate locally without requiring the account to exist in the Kerberos repository. This is done by use of authenticate with option to sp_modifylogin or sp_addlogin.

```
sp_addlogin nightlybatch, localpassword, ...
go
sp_modifylogin nightlybatch, 'authenticate with', 'ASE'
go
```

Example 2: Migrating users from Adaptive Server authentication to LDAP user authenication In this example, a phased approach for moving clients from local ASE authentication to LDAP user authentication is given.The LDAP directory server has been setup but has not yet been populated with all user accounts. A small population of users has agreed to a pilot program to test out LDAP for external authentication to Adaptive Server.

```
sp_configure 'enable ldap user auth', 1
go
```

This setting allows failover to ASE authentication when authentication with the LDAP directory server fails or the LDAP server is unavailable. Users without accounts in LDAP, failover to ASE authentication. The users in the pilot program are added to the LDAP directory server and may begin authenticating using the LDAP directory server.

A user can determine which authentication mechanism was used to authenticate with with the global variable *@@authmech*:

```
select @@authmech
```

As the pilot program proceeds and LDAP authentication is used, users in the pilot program can be required to only use LDAP authentication:

```
sp_maplogin loginame, 'authenticate with', 'ldap'
go
```

When the pilot program concludes and LDAP user authentication is required for all users, change the configuration parameter to level 2. Any logins set to authenticate with LDAP during the pilot program can be reset to the default value ANY. The logins will still be required to authenticate using LDAP because the configuration parameter is set to 2.

```
sp_configure 'enable ldap user auth', 2
go
sp_maplogin loginame, 'authenticate with', 'any'
go
```

Note If a login has authenticate with set to a specific authentication mechanism such as LDAP, Kerberos, PAM, or ASE, then that login may only use that mechanism for authentication. It is best utilized to set exceptions to the server-wide settings and to force a particular authentication mechanism to be used.

Using *sybmapname* to handle user principal names

The purpose of sybmapname is to convert external user principal names used in Kerberos environment to the name space of Adaptive Server user logins. sybmapname is a user customizable shared object that can map names given on its input buffer to a name suitable for Adaptive Server login on its output buffer.

The sybmapname shared object can be used to do the custom mapping between the user principal name and the Adaptive Server login name. This shared object is optionally loaded at server startup, and the function syb__map_name contained in the shared object is called after a successful Kerberos authentication and just before the user principal is mapped to a login in the syslogins table. It can be customized to suit the users needs. This function is useful when the user principal name and the login name to be mapped are not identical.

The customizable logic is the function:

```
syb__map_name(NAMEMAPTYPE *protocol, char *orig,
int origlen, char *mapped, int *mappedlen)
```

where:

- `NAMEMAPTYPE *protocol` refers to a structure reserved for usage of this function.
- `char *orig` is an input buffer that is not null terminated.
- `int origlen` is the input buffer length. It should be less than or equal to 255 characters.
- `char *mapped` is an output buffer that should not be null terminated.
- `int *mappedlen` is an output buffer length. It should be less than or equal to 30.

The function returns a value greater than 0 if the mapping succeeds, it returns a value of 0 if no mapping occurred, and it returns a value less than 0 when an error occurs in syb__map_name(). When an error occurs, a message displays in the Adaptive Server errorlog to report the mapping failure.

For example, to authenticate a Kerberos user on Adaptive Server, first you must configure Adaptive Server to use the Kerberos security mechanism. For information on Kerberos configuration, see the Adaptive Server *System Administration Guide* and Open Client/Server documentation, and the white paper titled “Configuring Kerberos for Sybase” the Sybase Web site at http://www.sybase.com/detail?id=1029260

A sample *sybmapname.c* file is located in *$SYBASE/$SYBASE_ASE/sample/server/sybmapname.c*

Note Modify the file to implement your logic using simple error free logic. Take precautions while coding as it may interfere with the proper running of Adaptive Server. See the “Precautions when using sybmapname” on page 17.

Build the shared object or DLL using the generic platform specific makefile supplied. The makefile may need to be modified to suit your platform specific settings.

Place the resulting shared object generated in a location specified in your *$LD_LIBRARY_PATH* on UNIX machines, and *PATH* variable on Windows machines. The file should have read and execute permissions for the ‘sybase’ operating system user.

Note Sybase recommends that only the ‘sybase’ user is allowed read and execute permissions, and that all other access should be denied.

Verifying your login to Adaptive Server using Kerberos authentication

To verify your login to Adaptive Server using Kerberos authentication, assume that:

- *$SYBASE* refers to your release and installation directory
- *$SYBASE_ASE* refers to the Adaptive Server version directory that contains your server binary.
- *$SYBASE_OCS* refers to the Open Client/Server version directory.

Example 1 If a client's principal name is `user@REALM`, and the corresponding entry in syslogins table is `user_REALM`, then sybmapname can be coded to accept the input string `user@realm` and to convert the input string to the output string `user_REALM`.

Example 2 If the client principal name is `user`, and the corresponding entry in syslogins table is `USER`, then sybmapname can be coded to accept the input string `user` and convert this string to uppercase string `USER`.

sybmapname is loaded by Adaptive Server at runtime and uses its logic to do the necessary mapping.

The following actions and output further illustrate the sybmapname function described in Example 2. The *sybmapname.c* file containing the customized definition for syb__map_name() should be compiled and built as a shared object (or DLL), and finally placed in the appropriate path location. Adaptive Server should be started with the Kerberos security mechanism enabled.

To initialize the TGT:

```
$ /krb5/bin/kinit johnd@public
Password for johnd@public:
$
```

To list the TGT:

```
$ /krb5/bin/klist
   Cache Type: Kerberos V5 credentials cache
   Cache Name: /krb5/tmp/cc/krb5cc_9781
Default principal: johnd@public
```

Login as "sa" and verify user login for `'johnd'`:

```
$ $SYBASE/$SYBASE_OCS/bin/isql -Usa -P
      -Ipwd`/interfaces
1>

1> sp_displaylogin johnd
2> go
```

```
No login with the specified name exists.
(return status = 1)

1> sp_displaylogin JOHND
2> go
Suid: 4
Loginame: JOHND
Fullname:
Default Database: master
Default Language:
Auto Login Script:
Configured Authorization:
Locked: NO
Password expiration interval: 0
Password expired: NO
Minimum password length: 6
Maximum failed logins: 0
Current failed login attempts:
Authenticate with: ANY
(return status = 0)
```

Successful Kerberos authentication, which maps lower case `johnd` to uppercase `JOHND` using the sybmapname utility and allows user `johnd` to login to Adaptive Server:

```
$ $SYBASE/$SYBASE_OCS/bin/isql -V -I`pwd`/interfaces
1>
```

Precautions when using *sybmapname*

You should be aware of the following issues when coding for sybmapname:

- Take care with the sample sybmapname.c program and any modifications to it. Avoid code that may segfault, that may call exit(), that may call system calls(), that may change UNIX signals, or make any blocking calls. Improper coding or calls may interfere with the Adaptive Server engine currently running.

  Note Coding errors in sybmapname are not Sybase's responsibility. The code is owned and supported by the user.

- Code defensively, check all pointers before de-referencing them, and avoid system calls. This must be a quick name filtering function.

- Avoid use of `'goto'` statements since, depending on the platform, these can have unexpected side effects.
- When multiple realms are used, you should take care to map the user principal names to a suitable login name to reflect the realm information. For example, if there are two users whose user principal names are `userA@REALMONE` and `userB@REALMTWO`. They are mapped to the login names `userA_REALMONE` and `userB_REALMTWO` respectively instead of `userA` or `userB`. This is for distinguishing the two users who belong to different realms.

MIT Kerberos client library support

Adaptive Server is now certified and tested with a newer version of MIT client libraries using MIT Kerberos client libraries. The MIT product level is MIT Kerberos version 1.4.x for UNIX platforms.

Additional fixes to enhance the reliability of Adaptive Server with Kerberos under stress conditions are available in Adaptive Server 12.5.4 release.

Expanded platform support for MIT Kerberos

Adaptive Server supports MIT Kerberos on the following new platform:

- HP-UX Itanium 64-bit

LDAP user authentication enhancements

LDAP user authentication has been enhanced in the following five ways:

- Support for secondary server lookup
- LDAP UA robustness enhancements
- Failover in case of errors while communicating with the LDAP server
- Password expiration related communication enhancements
- Tighter controls on login mapping

Configuring Adaptive Server for LDAP user authentication

LDAP user authentication allows client applications to send user name and password information to Adaptive Server for authentication by the LDAP server instead of syslogins. Authentication using the LDAP server allows you to use server-wide passwords instead of Adaptive Server or application-specific passwords. LDAP user authentication lets you simplify and centralize user administration.

LDAP user authentication works with directory servers that meet version 3 of the LDAP protocol standard, including Active Directory, iPlanet, and OpenLDAP Directory Server.

You can use two authentication algorithms with LDAP user authentication,. which differ in how they obtain a user's distinguished name (DN). The algorithms use either:

- Composed DN for authentication, available for Adaptive Server version 12.5.1 or later, or,
- Searched DN for authentication, available for Adaptive Server version 12.5.2 and later.

The primary data structure used with the LDAP protocol is the LDAP URL.

An LDAP URL specifies a set of objects or values on an LDAP server. Adaptive Server uses LDAP URLs to specify an LDAP server and search criteria to use to authenticate login requests.

The LDAP URL uses this syntax:

```
ldapurl::=ldap://host:port/node/?attributes?base | one | sub?filter
```

where:

- *host* – is the host name of the LDAP server.
- *port* – is the port number of the LDAP server.
- *node* – specifies the node in the object hierarchy at which to start the search.
- *attributes* – is a list of attributes to return in the result set. Each LDAP server may support a different list of attributes.
- base | one | sub – qualifies the search criteria. base specifies a search of the base node; one specifies a search of the base node and one sublevel below the base node; sub specifies a search of the base node and all node sublevels.

- filter – specifies the attribute or attributes to be authenticated. The filter can be simple, such as `uid=*`, or compound, such as `(uid=*)(ou=group)`.

Composed DN algorithm

The following steps describe the login sequence when you use the composed DN algorithm:

1. Open Client™ connects to an Adaptive Server listener port.
2. The Adaptive Server listener accepts the connection.
3. Open Client sends an internal login record
4. Adaptive Server reads the login record.
5. Adaptive Server binds to the LDAP server with a DN composed from the primary URL and the login name from the login record. This bind also uses the password from the login record.
6. The LDAP server authenticates the user, returning either a success or failure message.
7. If the primary URL specifies a search, then Adaptive Server sends the search request to the LDAP server.
8. The LDAP server returns the results of the search.
9. Adaptive Server accepts or rejects the login, based on the search results.

Searched DN algorithm

The following steps describe the login sequence when you use the searched DN algorithm:

1. Open Client connects to an Adaptive Server listener port.
2. The Adaptive Server listener accepts the connection.
3. Open Client sends an internal login record.
4. Adaptive Server reads the login record.
5. Adaptive Server binds to the LDAP server with a directory server access account.
6. The LDAP server authenticates the user, returning either a success or failure message.

The connection established in steps 5 and 6 may persist between authentication attempts from Adaptive Server to reuse connections to DN searches.

7 Adaptive Server sends search requests to LDAP server based on the login name from the login record and the DN lookup URL.

8 The LDAP server returns the results of the search.

9 Adaptive Server reads the results to obtain a value of attribute from the DN lookup URL.

10 Adaptive Server uses the value of attribute as the DN and the password from the login record to bind to the LDAP server.

11 The LDAP server authenticates the user, returning either a success or failure message.

12 If the primary URL specifies a search, Adaptive Server sends the search request to the LDAP server.

13 The LDAP server returns the results of the search.

14 Adaptive Server accepts or rejects the login, based on the search results.

Adaptive Server reports a generic login failure to the client if any of these authentication criteria are not met.

You may skip steps 12 and 13 by not specifying search criteria in the primary or secondary URL strings. When you do not specify criteria in the primary or secondary URL strings, the authentication completes, displaying the success or failure returned by step 11.

Secondary lookup server support

Adaptive Server version 12.5.4 provides uninterrupted support to Adaptive Server clients authenticated by an LDAP server. You can now specify a secondary LDAP lookup server to fail over from a primary LDAP server in the event of the LDAP server failure or planned downtime.

The health of the URL set is monitored through the following states:

- INITIAL – indicates that LDAP user authentication is not configured.
- RESET – indicates that the URL has been entered with Adaptive Server administrative commands.
- READY – indicates that the URL is ready to accept connections.

- ACTIVE – indicates that the URL has performed a successful LDAP user authentication.
- FAILED – indicates that there is a problem connecting to the LDAP server.
- SUSPENDED – indicates that the URL is in maintenance mode, and will not be used.

The following sequence of events describe the fail over and manual fail back:

1. The primary and secondary URL sets are configured and in a READY state.
2. The connections are authenticated using the primary server infrastructure.
3. The primary server fails, and its state is changed to FAILED.
4. Connections automatically begin authentication through the secondary server infrastructure.
5. The primary server is repaired and brought back online by an LDAP administrator. The primary LDAP server state is changed by an Adaptive Server administrator to READY.
6. New connections are authenticated using the primary server.

Note Once Adaptive Server has failed over to the secondary LDAP server, a database administrator must manually activate the primary LDAP server before it can be used again.

When Adaptive Server encounters errors connecting to an LDAP server, it retries the authentication three times. If the errors persist the LDAP server is marked as FAILED. See “Troubleshooting LDAP user authentication errors” on page 27 for information on the LDAP errors which force Adaptive Server to get into a retry loop.

Adaptive Server version 12.5.4 introduces the following new sp_ldapadmin options to support secondary lookup LDAP servers:

- To set the secondary DN lookup URL, enter:

```
sp_ldapadmin set_secondary_dn_lookup_url, <URL>
```

- To set the administrative access account for the secondary DN lookup URL, enter:

```
sp_ldapadmin set_secondary_access_acct, <DN>,
<password>
```

- To suspend the use of a primary or secondary URL for authentication, enter:

  ```
  sp_ldapadmin suspend, {primary | secondary}
  ```

- To activate the set of primary or secondary URLs for authentication, enter:

  ```
  sp_ldapadmin activate, {primary | secondary}
  ```

- To display details about the primary and secondary LDAP Server settings and status, enter:

  ```
  sp_ldapadmin list
  ```

  sp_ldapadmin list combines previous outputs from list_access_acct and list_urls. It has the following expected output for the primary and secondary servers:

  - Search URL
  - Distinguished Name Lookup URL
  - Access Account DN
  - Active [True | False]
  - Status [Ready | Active | Failed | Suspended | Reset]

Adaptive Server version 12.5.4 includes the following sp_ldapadmin option changes to support secondary servers.

- To display DN lookup URLs for the secondary server, enter:

  ```
  sp_ldapadmin list_urls
  ```

- To display the administrative account for the secondary DN lookup URL, enter:

  ```
  sp_ldapadmin list_access_acct
  ```

- To display new subcommands, enter:

  ```
  sp_ldapadmin [help | invalid sub-command]
  ```

LDAP server state transitions

The following tables list LDAP server state transitions when each of the sp_ldapadmin commands is executed.

Table 2-1 shows the state transitions when you execute sp_ldapadmin set_URL, where set_URL represents either of the following commands:

- set_dn_lookup_url
- set_primary_url
- set_secondary_dn_lookup_url
- set_secondary_url

Table 2-1: State transitions when sp_ldapadmin set_URL is executed

Initial state	**Final state**
INITIAL	RESET
RESET	RESET
READY	READY
ACTIVE	RESET
FAILED	RESET
SUSPENDED	RESET

Table 2-2 shows the state transitions when you execute sp_ldapadmin suspend.

Table 2-2: State transitions when sp_ldapadmin suspend is executed

Initial state	**Final state**
INITIAL	Error
RESET	SUSPENDED
READY	SUSPENDED
ACTIVE	SUSPENDED
FAILED	SUSPENDED
SUSPENDED	SUSPENDED

Table 2-3 shows the state transitions when you execute sp_ldapadmin activate.

Table 2-3: State transitions when sp_ldapadmin activate is executed

Initial state	**Final state**
INITIAL	Error
RESET	READY
READY	READY
ACTIVE	ACTIVE
FAILED	READY
SUSPENDED	READY

The following tables show the LDAP server state transitions carried out implicitly by Adaptive Server.

Table 2-4 shows the state transitions when Adaptive Server is restarted:

Table 2-4: State transitions when Adaptive Server is restarted

Initial state	Final state
INITIAL	INITIAL
RESET	RESET
READY	READY
ACTIVE	READY
FAILED	FAILED
SUSPENDED	SUSPENDED

Adaptive Server only attempts an LDAP login if the LDAP server is in a READY or ACTIVE state. Table 2-5 shows the state transitions:

Table 2-5: State transitions when an LDAP login succeeds

Initial state	Final state
READY	ACTIVE
ACTIVE	ACTIVE

Table 2-6 shows the state transitions when an LDAP login fails:

Table 2-6: State transitions when an LDAP login fails

Initial state	Final state
READY	FAILED
ACTIVE	FAILED

LDAP UA robustness enhancements

Adaptive Server version 12.5.4 introduces a number of new sp_ldapadmin options to enhance robustness.

Maximum ldapua native threads per engine

set_max_ldapua_native_threads sets the maximum number of native threads that can be running concurrently in an engine processing an LDAP authentication request.

```
sp_ldapadmin 'set_max_ldapua_native_threads', 'an integer'
```

The minimum value of set_max_ldapua_native_threads is 1. The maximum value is max native threads minus number of dump threads as specified using sp_configure.The default value is the same as the maximum value.

sp_configure ensures that max native threads is sufficient for set_max_ldapua_native_threads and the value of the configuration parameter number of dump threads.

LDAP request timeout set_timeout sets the time in milliseconds that Adaptive Server waits for a response from the LDAP server before abandoning the authentication request.

You can set this option by entering:

```
sp_ldapadmin, 'set_timeout', 'time_in_milli_seconds'
```

The default value for set_timeout is 10,000 milliseconds (10 seconds.) Valid values are between 1 and 3,600,000 (one hour.)

Abandon LDAP authentication when full set_abandon_ldapua_when_full allows you to seek alternative means of LDAP user authentication when the native threads per engine capacity is exceeded.

When no more threads are available, the request is abandoned if set_abandon_ldapua_when_full is set to true. If enable ldap user auth is set to 1, the client is authenticated using Adaptive Server syslogins. If enable ldap user auth is set to 2, the client login fails.

If set_abandon_ldapua_when_full is set to false, the authentication request is blocked until the LDAP descriptor can accept new authentication requests.

To set set_abandon_ldapua_when_full, enter:

```
sp_ldapadmin 'set_abandon_ldapua_when_full',
      'true | false'
```

The default value is false. Valid values are true and false.

LDAP descriptors per engine The login sequence of searched DN algorithm requires Adaptive Server to bind to the LDAP server using the access account before it can perform searches. Adaptive Server obtains an LDAP descriptor (handle) as a result of the bind. This descriptor is used for searching the DN of the login on the LDAP server.

In Adaptive Server versions earlier than 12.5.4, there was only one descriptor per engine. While this descriptor was being used to perform a search by an incoming connection, other connections waited for the descriptor to become available. Adaptive Server version 12.5.4 can open up to 20 descriptors per engine. This provides improved concurrency and login performance.

For information about the searched DN algorithm, see “Searched DN algorithm” on page 20.

Troubleshooting LDAP user authentication errors

Adaptive Server may experience the following transient errors when communicating with the LDAP server. These errors are generally resolved by trying the connection again. If the errors persist after three retry attempts, Adaptive Server marks the LDAP server as FAILED.

- LDAP_BUSY – server is busy.
- LDAP_CONNECT_ERROR – error during a connect.
- LDAP_LOCAL_ERROR – an error on the client side.
- LDAP_NO_MEMORY – cannot allocate memory on the client side.
- LDAP_OPERATIONS_ERROR – error on the server side.
- LDAP_OTHER – unknown error code.
- LDAP_ADMINLIMIT_EXCEEDED – a search has exceed a limit.
- LDAP_UNAVAILABLE – server cannot process the request.
- LDAP_UNWILLING_TO_PERFORM – server is not going to process the request.
- LDAP_LOOP_DETECT – a loop detected during a referral.
- LDAP_SERVER_DOWN – server is not reachable (connection fails).
- LDAP_TIMEOUT – LDAP API fails because operation does not complete in the user-specified amount of time.

Transient errors and a large number of simultaneous login requests could lead errorlog with a large number of repeated error messages. To increase the readability of the log, the following error message logging algorithm is used:

1. If a message is being logged for the first time, log it.
2. If the last time the message was logged was greater than 3 minutes:
    - Log the error message.
    - Log the number of times the message was repeated since the message was last printed.
    - Log the time in minutes elapsed since the message was printed.

Authentication failures arising from the following are not considered LDAP errors and are not conditions for retrying the authentication request.

- Bind failure due to bad password or an invalid distinguished name.

- A search after a successful bind that returns a result set of 0 or no attribute value.

Syntax errors found while parsing the URL are caught when an LDAP URL is set, and therefore do not fall in any of the above categories.

LDAP user authentication administration

The syntax of the sp_ldapadmin command is as follows:

sp_ldapadmin Usage: sp_ldapadmin command [, option1 [, option2]]

sp_ldapadmin commands include:

- `sp_ldapadmin 'set_primary_url', 'url'`
- `sp_ldapadmin 'set_secondary_url', 'url'`
- `sp_ldapadmin 'set_dn_lookup_url', 'url'`
- `sp_ldapadmin 'set_secondary_dn_lookup_url', 'url'`
- `sp_ldapadmin 'set_access_acct', 'distinguished name', 'password'`
- `sp_ldapadmin 'set_secondary_access_acct', 'distinguished name', 'password'`
- `sp_ldapadmin 'suspend', {'primary' | 'secondary'}`
- `sp_ldapadmin 'activate', {'primary' | 'secondary'}`
- `sp_ldapadmin 'list'`
- `sp_ldapadmin 'list_urls'`
- `sp_ldapadmin 'list_access_acct'`
- `sp_ldapadmin 'check_url', 'url'`
- `sp_ldapadmin 'check_login', 'name'`
- `sp_ldapadmin 'set_timeout', timeout_in_milli_seconds`
- `sp_ldapadmin 'set_max_ldapua_native_threads', max_ldapua_native_threads`
- `sp_ldapadmin 'set_abandon_ldapua_when_full', {true|false}`
- `sp_ldapadmin 'help'`

The following is an example output from a client administration session:

```
1> sp_configure 'enable ldap', 2
2> go

 Parameter Name                Default     Memory Used Config Value
         Run Value   Unit                 Type
------------------------------ ----------- ----------- ------------
         ----------- -------------------- ----------
enable ldap user auth                    0           0            2
                   2 not applicable       dynamic

(1 row affected)
```

```
1> sp_ldapadmin
'set_primary_url','ldap://primldap:30001/'
2> go
The URL 'ldap://primldap:30001/' is set for LDAP User
Authentication.
(return status = 0)

1> sp_ldapadmin 'set_dn_lookup_url',
'ldap://primldap:30001/dc=sybase,dc=com??sub?uid=*'
2> go
The URL
'ldap://primldap:30001/dc=sybase,dc=com??sub?uid=*'
is set for LDAP User Authentication.
(return status = 0)

1> sp_ldapadmin
'set_access_acct','cn=directorymanager,dc=sybase,
dc=com', 'primpassword'
2> go
The LDAP account distinguished name 'cn=directory
manager,dc=sybase,dc=com' is set for LDAP user
authentication.
(return status = 0)

1> sp_ldapadmin 'set_secondary_url',
'ldap://secldap:31001/'
2> go
The URL 'ldap://secldap:31001/' is set for LDAP User
Authentication.
(return status = 0)

1> sp_ldapadmin 'set_secondary_dn_lookup_url',
'ldap://secldap:31001//dc=sybase,dc=com??sub?uid=*'
2> go
The URL
'ldap://secldap:31001//dc=sybase,dc=com??sub?uid=*'
is set for LDAP User Authentication.
(return status = 0)
2> sp_ldapadmin 'set_secondary_access_acct',
'cn=Manager,dc=sybase,dc=com', 'secpassword'
3> go
The LDAP account distinguished name
'cn=Manager,dc=sybase,dc=com' is set for LDAP user
authentication.
(return status = 0)
1> sp_ldapadmin activate, primary
2> go
(return status = 0)

1> sp_ldapadmin activate, secondary
2> go
(return status = 0)


1> sp_ldapadmin 'list'
2> go
Primary:
      URL:                  'ldap://primldap:30001/'
      DN Lookup URL:
      'ldap://primldap:30001/dc=sybase,dc=com??sub?uid
      =*'
      Access Account:
      'cn=directory manager,dc=sybase,dc=com
      Active:               'TRUE'
      Status:               'READY'
Secondary:
      URL:                  'ldap://secldap:31001/'
      DN Lookup URL:
     'ldap://secldap:31001/dc=syase,dc=com??sub?uid=*'
      Access Account:     'cn=Manager,dc=sybase,dc=com'
      Active:               'TRUE'
      Status:               'READY'
Timeout value:            '-1'(10000) milliseconds
Maximum LDAPUA native threads per Engine: '49'
Abandon LDAP user authentication when full: 'false'
(return status = 0)

1> sp_ldapadmin 'list_urls'
2> go
```

```
Primary URL:          'ldap://primldap:30001/'
Secondary URL:        'ldap://secldap:31001/'
Distinguished Name Lookup URL:
'ldap://primldap:30001/dc=sybase,dc=com??sub?uid=*'
Secondary Distinguished Name Lookup URL:
'ldap://secldap:31001/dc=sybase,dc=com??sub?uid=*'
(return status = 0)

1> sp_ldapadmin 'list_access_acct'
2> go
Access Account DN:
'cn=directory manager,dc=sybase,dc=com'
Secondary Access Account DN:
'cn=Manager,dc=sybase,dc=com'
(return status = 0)
```

LDAP user authentication tuning

You can configure and tune Adaptive Server options based on load of incoming connections and the Adaptive Server-LDAP server infrastructure. You can configure the following two options based on the number of simultaneous incoming requests:

- Use sp_configure to set max native threads, which indicates the number of native threads per engine.
- Use sp_ldapadmin to configure max_ldapua_native_threads, which indicates the number of LDAP user authentication native threads per engine.

Configure the following option based on the network and the health of the Adaptive Server/LDAP server infrastructure:

- Use sp_ldapadmin to configure set_timeout which indicates the LDAP server bind and search timeouts.

Configure the following option to specify Adaptive Server behavior when incoming connections have consumed max_ldapua_native_threads:

- Use sp_ldapadmin to configure set_abandon_ldapua_when_full.

LDAP user authentication password information changes

There are two new LDAP user authentication-related informational messages that Adaptive Server obtains from the LDAP server and passes on to the client:

- If you log in to an Adaptive Server using an LDAP authentication mechanism with an LDAP user authentication password that is about to expire, the following message displays:

  ```
  Your password will expire in <number> days.
  ```

- If you attempt to log in to Adaptive Server using an LDAP authentication mechanism after the LDAP server administrator resets your password or after your LDAP server password has expired, you will get a message 4002:

  ```
  Login failed
  ```

  If auditing is enabled and errors auditing option is turned on:

  ```
  sp_audit 'errors', 'all', 'all', 'on'
  ```

  a 4099 message is sent to the audit log. The 4099 message reads:

  ```
  Your LDAP password has expired.
  ```

Note Configure your LDAP server to give this additional information. Additionally, Adaptive Server must support the transmission of LDAP password controls to an LDAP client.

Adding tighter controls on login mapping

Use sp_maplogin to map external client names to local Adaptive Server logins. It is used to map users that are authenticated with LDAP or PAM to the local Adaptive Server login.

To map a user authenticated with Kerberos, use sybmapname instead of sp_maplogin. See instructions and examples in “Using sybmapname to handle user principal names” on page 14.

Only users with sso_role can create or modify login mappings using sp_maplogin.

In Adaptive Server version 12.5.4, controls have been added to sp_maplogin to avoid conflicts between an authentication mechanism setting for a login and a mapping that uses the login. Potential mapping conflicts are detected by the stored procedures sp_maplogin, sp_modifylogin, or sp_addlogin.

The tighter controls no longer permit a map:

- From one Adaptive Server login name to another login name
- From an external name that already exists as a local login
- To a nonexistent login name

Additionally, when the authentication mechanism is specified with a mapping, the mechanism is checked with the authentication mechanism set in the target login.

If a target login's authentication mechanism restricts the login to use a particular authentication mechanism, then the mechanism specified with the mapping must match either that specified for the login or match ANY authentication mechanism.

When sp_maplogin detects that a conflict exists, sp_maplogin fails and reports an error to identify the conflict.

Similarly, sp_modifylogin and sp_addlogin check for an existing mapping that may conflict with the authenticate with option for the user login.

When sp_modifylogin or sp_addlogin detect a conflict, then an error is reported to identify any conflicts with a login mapping.

Example 1: Mapping an LDAP user to Adaptive Server "sa" login A company has adopted LDAP as their repository for all user accounts and has a security policy that requires LDAP authentication of all users including database administrators, `"adminA"` and `"adminB",` who may manage hundreds of Adaptive Servers. Auditing is enabled and login events are recorded in the audit trail.

To map these administrator accounts to "sa", enter:

```
sp_maplogin LDAP, 'adminA', 'sa'
go
sp_maplogin LDAP, 'adminB', 'sa'
go
```

Use enable ldap user auth to require all users to authenticate using LDAP authentication:

```
sp_configure 'enable ldap user auth', 2
go
```

When 'adminA' authenticates during login to Adaptive Server, the distinguished name associated with 'adminA' rather than only “sa” is recorded in the login audit event. This allows each individual performing an action to be identified in the audit trail.

Because 'adminA' and 'adminB' password is set in the LDAP server, there is no need to maintain the “sa” password on all Adaptive Servers being managed.

This example also allows different external identities and passwords to be used for authentication, while their actions within Adaptive Server still require the special privileges associated with “sa” account.

Example 2: Using both PAM and LDAP to map users to application logins A company has adopted both PAM and LDAP authentication but for different purposes. The company security policy defines LDAP as the authentication mechanism for general user accounts, and PAM for special users such as a middle-tier application. A middle-tier application may establish a pool of connections to Adaptive Server to handle requests on behalf of users of the middle-tier application.

Configure Adaptive Server for both LDAP and PAM user authentication:

```
sp_configure 'enable ldap user auth', 2
go
sp_configure 'enable pam user auth', 2
go
```

Establish an Adaptive Server login appX locally with permissions that are appropriate for the middle-tier application:

```
sp_addlogin 'appX', password
go
sp_modifylogin appX, 'authenticate with', PAM
go
```

Instead of hard-coding a simple password in appX and maintaining the password consistently in several different Adaptive Servers, a custom PAM module is developed to authenticate the application in a centralized repository using additional facts to verify the middle tier application.

Client application login appY requires LDAP authentication of the user with its LDAP identity and password. Use sp_maplogin to map all LDAP authenticated users to login appY.

```
sp_addlogin 'appY', password
go
sp_maplogin LDAP, NULL, 'appY'
go
```

Users of appY are authenticated with their company identity and password, then mapped to a local Adaptive Server login appY to execute database actions. Authentication has occurred with the identity of the LDAP user, which is recorded in the audit trail, and executes with permissions appropriate to the application login appY.

SSL support

Adaptive Server version 12.5.4 supports SSL on the following new platforms:

- HP-IA64
- Linux64

PAM support

Adaptive Server version 12.5.4 supports PAM on:

- HP-IA64
- Macintosh OS X

Updates to encrypted columns

Adaptive Server version 12.5.4 includes the following encrypted columns enhancements. For information about the first version of the encrypted columns feature which was part of version 12.5.3a, see the *Encrypted Columns Guide.*

Referential integrity with encrypted columns

You can establish referential integrity between two encrypted columns when:

- You use the same key to encrypt the referenced and referencing columns.

- The key used to encrypt the columns specifies an initialization vector NULL and a random pad NULL.

Referential checks are efficient as they are performed on encrypted values.

alter table and encrypted columns

You cannot use alter table to encrypt or decrypt a column belonging to a clustered or placement index. To encrypt or decrypt this column, drop the index, alter the column and re-create the index.

You cannot use alter table to decrypt a column if the table has a trigger defined. To decrypt the column, drop the trigger, alter the column and then re-create the trigger.

sp_help and encrypted columns

sp_help displays information about encryption keys. When a key name is specified as the parameter to sp_help, the command lists the key's name, owner, object type, and creation date.

sp_helprotect and encrypted columns

sp_helprotect reports new information on encrypted columns, encryption keys, and users as follows:

- Tables and columns – reports who has been granted decrypt permission and on which columns.
- Encryption keys – reports who has been granted select permission.
- Users – indicates users who have been granted create encryption key permission.

Password complexity and login options

Adaptive Server version 12.5.4 introduces a combination of new and old mechanisms that allows you to establish rules about passwords for new logins or for passwords that are being reset.

As with earlier versions of Adaptive Server, you can:

- Require that passwords contain at least one digit.
- Require that passwords have at least a minimum length.
- Set a password expiration period.
- Require that a login will get locked out after a certain number of failed attempts to log in.
- Set an individual login to have its own rules for digits, minimum length, and login failures. The per-login rules override the global logins for that user.

However, Adaptive Server version 12.5.4 you can also:

- Specify that the login name should not be a substring of the password.
- Set a minimum number of special characters for the password.
- Set a minimum number of alphabetic characters for the password
- Set a minimum number of upper case letters for the password.
- Set a minimum number of lower case letters for the password.
- Specify that the password must be reset at first log on.
- Set a minimum number of digits for the password.
- Set password expiration warning interval.

You can set each of these new options in the Adaptive Server plug-in, or you can use a new stored procedure:

```
sp_passwordpolicy 'set', option, value
```

For information about each new option and its valid values, see New password complexity checks below.

Setting new password complexity options creates a row for each option in the sysattributes table. As long as the row for the new option exists, precedence checking uses the new option value and ignores any older corresponding option values.

To return to earlier versions of password rules, either unselect the password complexity options using the Adaptive Server plug-in, or use:

```
sp_passwordpolicy 'clear’, option
```

The new password complexity options also have cross checks. For example, if the sum of the min lower case in password and min upper case in password is greater than the min alpha in password, a warning message displays.

New password complexity checks

The options that support password complexity checks are discussed in this section. You can use these options in a new stored procedure interface; their values are stored in the master.dbo.sysattributes table.

To turn these password options off, use sp_passwordpolicy. To turn off an option for an individual option, enter:

```
sp_passwordpolicy 'clear', option
```

To turn off password policy options for all passwords, enter:

```
sp_passwordpolicy 'clear'
```

Disallowing simple passwords

disallow simple password checks to see if the password contains the login name as a substring. It can be set to:

- 0 – turns off the option and simple passwords are allowed.
- 1 – turns the option on, and disallows simple passwords.

By default, this option is turned off and this check is not applied to passwords.

To set this option, enter:

```
sp_passwordpolicy 'set', 'disallow simple passwords', 1
```

When you disallow simple passwords, you cannot use your login name as a substring in your password. You must set it to something complex. For example:

```
sp_password 'old_complex_password', BHotAcha789, johnd
```

The login `johnd` now has a password of `BHotAcha789`, which does not contain the login name as a substring.

However, if you change the login password entering:

```
sp_password 'old_complex_password', johnd123, johnd
```

the login johnd is now a substring of the new password johnd123, and this command fails.

Custom password complexity checks

Adaptive Server version 12.5.4 allows you to custom configure password checking above and beyond the password complexity rules using two stored procedures:

- sp_extrapwdchecks
- sp_cleanpwdchecks

These are system security officer provided stored procedures defined and located in the master database. These custom stored procedures are automatically invoked during the Adaptive Server password complexity checks, and when dropping a login, respectively. See “Enabling custom password checks” on page 50 for an example of how to create these custom stored procedures.

Specifying a minimum number of digits in a password

Use min digits in password to specify the minimum number of digits in a password. Valid values are:

- 0 through 16 – the minimum number of digits that must exist in a password.
- -1 – the password cannot contain digits.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set this option, enter:

```
sp_passwordpolicy 'set', 'min digits in password',
      number
```

For example, if you have set min digits in password to 4, you must have at least four digits in your password. To satisfy this requirement, you can add a login johnd, with a complex password SEcret6789 by entering:

```
sp_addlogin 'johnd', 'SEcret6789'
```

However, if you enter:

```
sp_addlogin 'johnd', 'secret123'
```

the command fails because the minimum number of digits allowed is 4.

Specifying a minimum number of alphabetic characters in a password

min alpha in password specifies the minimum number of alphabetic characters allowed in a password. This value must be at least the sum of minimum number of uppercase characters and minimum number of lowercase characters.

Valid values are:

- 0 through 16 – the number of alphabetic characters required to be in the password.
- -1 – the password cannot contain alphabetic characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of alphabetic characters in a password, enter:

```
sp_passwordpolicy 'set', 'min alpha in password',
      number
```

For example, if you have set min alpha in password to 4, you must have at least 4 alphabetic characters in your password. To satisfy this requirement, you can set the password for new login `johnd` to `SECret123456`, by entering:

```
sp_addlogin 'johnd', SECret123456
```

However, if you attempt to create the new login:

```
sp_addlogin 'johnd', 'sec123456'
```

and the minimum number of alphabetic characters is still 4, the command fails.

Specifying a minimum number of special characters in a password

min special char in password specifies the minimum number of special characters for a password. Valid values are:

- 0 through 16 – the minimum number of special characters required for a password.
- -1 – the password cannot contain special characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of special characters in a password, enter:

```
sp_passwordpolicy 'set', 'min special char in
      password',number
```

For example, if you set min special char in password to 3, you can create a new login johnd with the password abcDE1&#$, by entering:

```
sp_addlogin, 'johnd', 'abcDE1&#$'
```

However, if you set the new login johnd to have a password of abcDE1#:

```
sp_addlogin, 'johnd', 'abcDE1#'
```

and the number of special characters is still set to 3, the command fails.

Specifying a minimum number of uppercase letters in a password

min upper char in password allows you to set the minimum number of uppercase letters for a password. Valid values are:

- 0 through 16 – the number of uppercase letters required for a password.
- -1 – the password cannot contain uppercase characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of uppercase characters allowed in a password, enter:

```
sp_passwordpolicy 'set', 'min upper char in password'’
      number
```

For example, if you set min upper char in password to 3, you can create a login johnd with the password abCDE1#, by entering:

```
sp_addlogin 'johnd', 'abCDE1#'
```

However, if you attempt to add the following login:

```
sp_addlogin 'johnd', 'abcDE1#'
```

and the minimum number of uppercase letters is set to 3, the command fails.

Specifying a minimum number of lowercase letters in a password

min lower char in password sets the minimum number of lowercase letters for a password. Valid values are:

- 0 through 16 – indicates the number of lowercase letters required for a password.

- -1 – indicates that the password cannot contain lowercase characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of lowercase letters in a password, enter:

```
sp_passwordpolicy 'set', 'min lower char in password',
      number
```

For example, if you set min lower char in password to 3, you can create a login `johnd` with the password `abcdEF1#`, by entering:

```
sp_addlogin 'johnd', 'abcdEF1#'
```

However, if you attempt to add the following login:

```
sp_addlogin 'johnd', 'abCDE#1'
```

and the minimum number of lowercase letters allowed is 3, the command fails.

Specifying the minimum password length

minimum password length sets the minimum password length. You can set a minimum password length from 0 – 30. The value you specify with must be at least the sum of all other minimum requirements.

For example, minimum password length must be set to at least 10 if you have set:

minimum digits in password to 3
minimum special characters in password to 2
minimum uppercase characters in password to 2
minimum lowercase characters in password to 3

In this example, if the password length is less than 10, a warning message displays, but the setting of the password policy option succeeds.

To set the minimum password length, enter:

```
sp_passwordpolicy 'set', 'minimum password length',
      number
```

For example, if you set minimum password length to 6, you can create a new password for login `johnd` with six characters by entering:

```
sp_password 'old_complex_password', 'ABcd12$%', 'johnd'
```

However, if you attempt to change your password as follows:

```
sp_password 'old_complex_password', 'joh12', 'johnd'
```

and the minimum password length is 6, the command fails.

Specifying the password expiration

password expiration specifies the number of days a password can exist before it expires. You specify this value on a global basis. Valid values include:

- 0 – the password will never expire.
- 1 through 32767 – the number of days the password can exist without expiring.

By default, this password complexity option is turned off and this check is not applied to passwords.

To specify the password expiration date, enter:

```
sp_passwordpolicy 'set', 'password expiration', number
```

Specifying the password expiration warning interval

password exp warn interval indicates the number of days before a password expires that the password expiration warning messages displays. These messages display with every successful login until the password is changed or it expires. This value must be less than or equal to the password expiration.

Valid values are 0 – 365. This option is turned off by default.

To specify the password expiration warning interval, enter:

```
sp_passwordpolicy 'set', 'password exp warn interval',
      number
```

Specifying the number of failed logins allowed

maximum failed logins specifies the maximum number of failed logins that can occur before the login is locked. You specify this value globally. Valid values are:

- 0 – logins are never locked, regardless of the number of failed login attempts.
- 1 through 32767 – the number of failed logins that can occur before the login is locked.

By default, this value is turned off and this check is not applied to passwords.

To set the number of failed logins allowed, enter:

```
sp_passwordpolicy 'set', 'maximum failed logins',
      number
```

Resetting the password at first login

expire login changes the login status to expired when a System Security Officer creates or resets a login. The login is then required to change the password on the first login. Valid values are:

- 0 – new or reset logins will not expire.
- 1 – new or reset logins do expire; reset your password at the first login.

By default, this value is turned off and this check is not applied to passwords.

To set the option to require a change of password at first login, enter:

```
sp_password policy 'set', 'expire login', [ 1 | 0 ]
```

Password complexity option cross-checks

For some password complexity options, you must be aware of the interaction between options.

- minimum password length must be at least the sum of min digits in password, min alpha in password, min special characters in password.
- min alpha in password must be at least the sum of min upper char in password and min lower char in password.
- systemwide password expiration must be greater than password exp warn interval.

For the purpose of the above cross-checks, if Adaptive Server encounters a password complexity option value of -1, it interprets that as a value of 0. If any of the options is not set, Adaptive Server interprets the option value to be 0 as well.

Adaptive Server prints warnings for each new password complexity option that fails to satisfy the cross-checks. Option setting, however, is successful.

Setting old and new password complexity checks

Table 2-7: Old and new password complexity checks

Password checks and policies for Adaptive Server authentication	Existing configuration parameters specified using sp_configure	New password complexity options specified using sp_passwordpolicy	Existing per-login overrides specified using sp_modifylogin
Password expiration	system-wide password expiration	system-wide password expiration	password expiration
Digits in password	check password for digit	min digits in password	N/A
Alphabetic characters in password	N/A	min alpha in password	N/A
Password length	minimum password length	minimum password length	min passwd length
Failed logins lockout	maximum failed logins	maximum failed logins	max failed_logins
Disallow simple passwords	N/A	disallow simple passwords	N/A
Special characters in password	N/A	min special char in password	N/A
Uppercase letters in password	N/A	min upper char in password	N/A
Lowercase letters in password	N/A	min lower char in password	N/A
Password expiration warning interval	N/A	password exp warn interval	N/A
Resetting your password at first login	N/A	expire login	N/A
Custom password complexity checks	N/A	N/A	N/A

You can set the password complexity options at:

- The login level using sp_addlogin or sp_modifylogin.
- The global level using the new sp_passwordpolicy or sp_configure.

Because you can set password configuration options on a global and per-login basis, and using old and new parameters, it is important to know the order of precedence in which the password options will be applied.

When applying password options: Adaptive Server looks first at the existing per-login parameters, then it looks at the new password complexity options, and then it looks at the existing global password options.

Example 1 If you enter:

```
sp_addlogin @login_name = 'johnd',
      @passwd = 'complex_password',
      @minpwdlen = 6
```

you have set the minimum password length for `johnd` to 6.

If you then enter the following existing global options for login `johnd`:

```
sp_configure 'minimum password length', 8
sp_configure 'check password for digit', 'true'
sp_passwordpolicy 'set', 'min digits in password', 2
```

you have created two minimum password length requirements for login `johnd`, and you have also set restrictions about digits in the password.

If you then try to create a password for login `johnd` as follows:

```
sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd123', @login_name = 'johnd'
```

Adaptive Server checks the password in the following order:

1. Per-login existing options check: minimum password length must be greater than 6. This is true and the check passes.
2. New options: minimum digits in password must be greater than 2. This is true and the check passes.
3. Existing global options: minimum password length specified here is not checked because there is already a per-login check for the login `johnd`.
4. The check password for digit option is redundant because it is already checked when the minimum number of digits is turned on and set to 2.

Once these checks have been performed in the designated sequence, and the new password for login `johnd` passes these checks, the new password is successfully created.

Example 2 If, for the same login, you enter:

```
sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd', @login_name = 'johnd'
```

Adaptive Server first checks the per-login existing options, and determines the minimum password length is set to 6, but that you have attempted to create a password with only 4 characters. The check fails, and Adaptive Server prints an error message. Once one password complexity check fails, no additional options are checked.

Example 3 If you attempt to create a new login with the following password configuration options:

```
sp_addlogin @login_name = 'johnd', @passwd =
'complex_password', @minpwdlen = 4
```

this sets the minimum password length for login johnd to 4. This is a per-login, existing option. If you then add:

```
sp_passwordpolicy 'set', 'min digits in password', 1
```

you have created a global requirement that the minimum number of digits for a password must be 1.

If you then attempt to create the password for login johnd as follows:

```
sp_password @caller_password = 'old_complex_password',
@ new_password = 'abcde', @login_name = 'johnd'
```

Adaptive Server performs the checks in the following order:

1. Per-login existing options check: the minimum password length of a new password is 4. The password abcde is greater than 4, so this check passes.
2. New global requirement check: the minimum digits in a password is set to 1, globally. This check fails.

Adaptive Server does not create a new password and prints an error message.

To create a new password, all the checks must pass.

Stored procedures for password complexity

Adaptive Server includes several new stored procedures to help configure password complexity.

sp_extrapwdchecks

sp_extrapwdchecks is a new custom stored procedure that can contain user-defined logic for password complexity checks. You can configure sp_extrapwdchecks according to your security needs. Install sp_extrapwdchecks in the master database.

```
sp_extrapwdchecks caller_password, new_password, login_name
```

where:

- *caller_password* specifies the current password.
- *new_password* specifies the new password being set.

- *login_name* specifies the login name associated with the password being changed or added.

sp_extrapasswordchecks must use raiserror to signal a failure to Adaptive Server. The error message for this failure should be added in Adaptive Server using sp_addmessage.

sp_cleanpwdchecks

sp_cleanpwdchecks is a new custom stored procedure that allows you to define when and how to remove login and password-related attributes stored in user-defined tables. sp_cleanpwdchecks is user defined, and is dynamically called in the master database when you drop a login.

```
sp_cleanpwdchecks, login_name
```

where:

- *login_name* – specifies the login name of the cleanup to be performed.

sp_passwordpolicy

sp_passwordpolicy is an interface that a user with a sso_role can use to specify, remove, and list new password complexity options. This information is stored in the master.dbo.sysattributes table.

```
sp_passwordpolicy {set | clear | list }, policy_option, option_value
```

where:

- set – sets a value to an option.
- clear – deletes the row for the option specified in the master.dbo.sysattributes table. If there is no policy option specified, clear deletes all the option rows in the sysattributes table.
- list – lists the values of the options specified.
- *policy_option* – the option parameter on which to perform an operation. Valid options are:
  - disallow simple passwords – a value of 1 turns this option on, and a value of 0 turns it off.
  - min digits in password – indicates the minimum number of digits to be allowed in a password.
  - min alpha in password – indicates the minimum number of alphabetic characters in a password.

    - min special char in password – indicates the minimum number of special characters allowed in a password.
    - min upper char in password – indicates the minimum number of upper case characters allowed in a password
    - min lower char in password – indicates the minimum number of lower case characters allowed in a password.
    - systemwide password expiration – indicates the system wide password expiration in days.
    - password exp warn interval – indicates the password expiration warning interval in days.
    - minimum password length – sets the minimum length of the password.
    - maximum failed logins – sets the maximum number of failed logins allowed in a session before the account is locked.
    - expire login – specifies that a login status changes to expired status when you create or reset your login. You are required to change your password on your first login.
- *option_value* – is the value for the *policy_option*.

Example 1

To set a password expiration warning interval to seven days before the password expires, enter:

```
sp_passwordpolicy 'set',
      'password exp warn interval', 7
```

Example 2

To list the option for minimum number of special characters, enter:

```
sp_passwordpolicy 'list',
      'min special char in password'
```

Example 3

To reset disallow simple password to the default value, enter:

```
sp_passwordpolicy 'clear', 'disallow simple passwords'
```

Changed stored procedures

The following stored procedures have been modified for password complexity implementation:

- sp_addlogin and sp_password – invoke the new password complexity checks.
- sp_droplogin – If sp_cleanpwdchecks is present in the master database, it is executed after dropping a login with sp_droplogin.

- sp_displaylogin – New sp_passwordpolicy security options are taken into consideration when displaying login information related to password expiration, maximum failed logins, and password length.

Enabling custom password checks

Adaptive Server version 12.5.4 allows a system security officer to write user-defined stored procedures that enable custom password checks.

For example, to implement password history checks, create a new user table to store password histories:

```
create table pwdhistory
(
      name varchar(30)not null,  -- Login name.
      passwordvarbinary(30)not null,  -- old password.
      pwdate datetime not null,  -- datetime changed.
      changedby varchar(30)not null  -- Who changed.
)
go
```

This user defined stored procedure can be called when specifying a new password to save it in an encrypted form in the pwdhistory table:

```
create proc sp_extrapwdchecks
(
@caller_password varchar(30), --the current password of caller
@new_password    varchar(30),-- the new password of the target acct
@loginame        varchar(30)-- user to change password on
)
as

begin
declare @current_time    datetime,
        @encrypted_pwd   varbinary(30),
        @changedby       varchar(30),
        @cutoffdate      datetime


select @changedby = suser_name()

-- Change this line according to your installation.
-- This keeps history of 12 months only.
select @current_time = getdate(),
       @cutoffdate = dateadd(month,-12,getdate())
```

```
select @encrypted_pwd = internal_encrypt(@new_password)

delete master..pwdhistory
    where  name = @loginame
    and    pwdate < @cutoffdate

if not exists ( select 1 from master..pwdhistory
                where name = @loginame
                and   password = @encrypted_pwd )
begin
      insert master..pwdhistory
      select @loginame, internal_encrypt(@caller_password),
             @current_time, @changedby
      return (0)
end
else
begin
      raiserror 22001  --user defined error message
      return (1)
end
end
go
```

Use sp_addmessage to add the user-defined message 22001. A raiserror 22001 indicates a custom password complexity check error and leads to a failure of sp_addlogin or sp_password.

The following user defined stored procedure can be used for clean up purposes after adding history using sp_extrapwdchecks.

```
create proc sp_cleanpwdchecks
(
            @loginame      varchar(30)
                        -- user to change password on
)
as
begin

delete master..pwdhistory
where name = @loginame
end
      go
```

Once the two procedures above are defined and installed in the master database, they are called dynamically during the password complexity checks.

DDLGen support

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 -N*username* instead of -N%:

```
ddlgen -Ulogin -Ppassword -Sserver:port -TL -Nusername
```

If server-wide password complexity options have been specified for the login or logins, all sp_addlogin and sp_modifylogin DDL statements are generated first, followed by DDL statements for the password complexity options.

This example generates DDL for the login “george” on a machine named HARBOR using port 1955:

```
ddlgen -Uroy -Proy123 -SHARBOR:1955 -TL -Ngeorge
```

Adaptive Server Plug-in support

The Adaptive Server Plug-in supports GUI administration for password complexity. For more information, see Chapter 8, “Adaptive Server Plug-in Changes.”

Exporting set options from a login trigger

Adaptive Server version 12.5.4 enables options set inside login triggers to remain valid for the entire user session.

The following set options are automatically exported:

- showplan
- arithabort [overflow | numeric_truncation]
- arithignore [overflow]
- colnames

- format
- statistics io
- procid
- rowcount
- altnames
- nocount
- quoted_identifier
- forceplan
- fmtonly
- close on endtran
- fipsflagger
- self_recursion
- ansinull
- dup_in_subquery
- or_strategy
- flushmessage
- ansi_permissions
- string_rtruncation
- prefetch
- triggers
- replication
- sort_resources
- transactional_rpc
- cis_rpc_handling
- strict_dtm_enforcement
- raw_object_serialization
- textptr_parameters
- sort_merge

- remote_indexes
- explicit_transaction_required
- statement_cache
- command_status_reporting
- proc_return_status
- proc_output_params

Setting global login triggers

Adaptive Server version 12.5.4 provides the ability to set a new global login trigger. Use sp_logintrigger to set a global login trigger, that is executed at each user login. If you want to take user specific actions, then you should set a user specific login trigger using sp_modifylogin or sp_addlogin.

sp_logintrigger

Description

sp_logintrigger is the new procedure that can be used to set and display the global login trigger. This global login trigger has the same characteristics as a personal login script. It is executed before any personal login script for every user that tries to log in, including system administrators and security officers.

Syntax

sp_logintrigger <*global login trigger name*>

Parameters

global login trigger name

is the name of the global login trigger.

If no parameter is included, sp_logintrigger displays the current login trigger status and name if it exists, and no rows if there is no global login trigger defined.

Examples

Example 1 To set a global login trigger using sp_logintrigger:

```
sp_logintrigger 'master.dbo.myproc'
```

Example 2 To view an updated global login trigger:

```
1> sp_logintrigger
2> go
```

```
Global login trigger                Status
---------------------------------- -------
sybsystemprocs.dbo.myproc           Enabled

(1 row affected)
(return status = 0
```

Example 3 When a global login trigger does not exist:

```
1> sp_logintrigger
2> go
Global login trigger Status
-------------------- ------
(0 rows affected)
```

Example 4 To delete a global login trigger that was specified earlier with sp_logintrigger, enter:

```
sp_logintrigger 'drop'
```

Usage

- A new global variable *@@logintrigger* is used to find out if a global login trigger is defined and enabled.
- There is a difference between this global login and the private login script. This global login trigger is stored by name in sysattributes, while the private login script is stored only by object ID.

Permissions

Everyone can execute sp_logintrigger to display the current global login trigger. To set a new login trigger, SSO role is required.

CHAPTER 3 Archive Database Access

Archive database access provides a database administrator with the capability to validate or selectively recover data from a database dump (an “archive”) by making the dump appear as if it were a traditional read-only database; this type of database is called an “archive database.”

Unlike a traditional database, an archive database uses the actual database dump as its main disk storage device, with a minimum amount of traditional storage to represent new or modified pages that result from the recovery of the database dump. A database dump already contains the images of many (if not most) of the database pages, therefore, an archive database can be loaded without having to use Backup Server to transfer pages from the archive to traditional database storage. Consequently, the load is significantly faster than a traditional database.

Overview

Archive database access enables a variety of operations to be performed directly on a database dump.

The amount of storage needed for a traditional database load must be equal to or greater than the size of the source database; the loading of the database dump using Backup Server involves copying pages from the database dump into the storage that has been set aside for the traditional database.

By contrast, you can create an archive database using a minimal amount of traditional disk storage. When you load an archive database, the pages residing in the database dump are not copied by the Backup Server. Instead, Adaptive Server creates a map that represents a "logical-to-virtual" mapping of the pages within the archive. This significantly decreases the amount of time required to view the data in a database dump, and reduces the storage requirement for loading the dump.

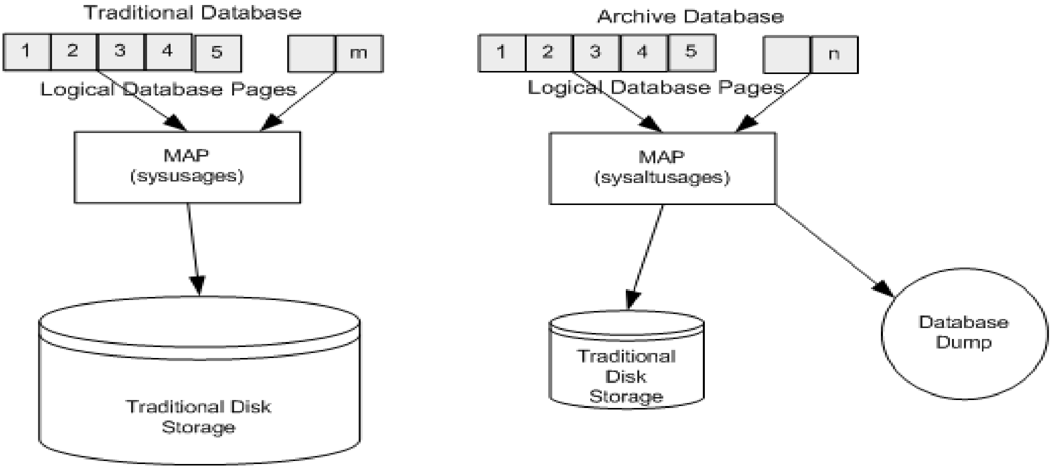
An archive database does not have to be a complete copy of the original database. Depending on the optimization used when dumping the database using sp_dumpoptimize, an archive database may be fully populated (every page in the database is in the database dump), or partially populated (only allocated pages are stored in the database dump).

Because the database dump is presented as a (read-only) database, a database administrator can query it using familiar tools and techniques such as:

- Running database consistency checks on the most recent copy of a dump made from a production database. These checks can be off-loaded to a different server to avoid resource contention in the production environment. If resources are not a concern, the archive can be directly checked in the same server in which it was created. Verification on the archive provides the assurance needed prior to performing a restore operation.
- If the integrity of a database dump is in question, loading it into an archive database can be a quick test for success and therefore a good tool to identify the appropriate database dump that should be used to restore a traditional database.
- Object-level restoration from the database dump. Lost data is recovered using select into to copy the to-be-restored rows from the table within the archive database. The select into operation can be performed either directly in the server hosting the archive database, or by using Component Integration Services proxy tables if the archive database is available on a different server than that of the object requiring restoration.

The following figure represents the differences between an archive database and a traditional database structure.

Figure 3-1: Archive database components



Components of an archive database

An archive database is made up of three components working together to create the illusion that a database dump is functioning as a traditional database. These components are:

- The database dump (the archive)
- Traditional disk storage used to store the modified pages section
- The scratch database that hosts the sysaltusages table

The database dump

The database dump is used as a repository for most unmodified pages.

The database dump is read-only. You cannot make changes to the database dump. Any changes you make to the data within the dump are stored in the modified pages section.

Adaptive Server sees the database dump and its stripes as database devices that are usable only by the archive database.

The modified pages section

Database dumps reflect a snapshot of a database at a given moment in time. The archive database that represents a database dump is read-only. No user transactions are allowed. Although the archive database is read-only, some modifications are allowed. For example:

- You can run recovery in order to make the archive database consistent.
- dbcc commands that perform fixes are allowed so that fixed versions of tables can be restored.

These modified and newly-allocated database pages cannot be stored within the database dump and its stripes, therefore an archive database requires some traditional database storage. This disk space is referred to as the modified pages section, and you can allocate it using the create archive database and alter database commands.

The scratch database and the sysaltusages table

The *sysaltusages* table

The sysaltusages system table is a new data-only-locked table that is used to map page numbers in an archive database to the actual page within either the database dump and its stripes or the modified pages section. However, unlike the sysusages table in a traditional database, the sysaltusages table does not map every logical page in the database. sysaltusages maps:

- Pages that have been stored in a database dump
- Pages that have been modified and therefore relocated to the modified pages section

sysaltusages has the following columns:

```
dbid    location    lstart    start    size    vstart    vdevno    segmap
```

where:

- dbid – is the database ID of the archive database.
- location – is the location of the archive database segment where the physically contiguous block of pages resides.

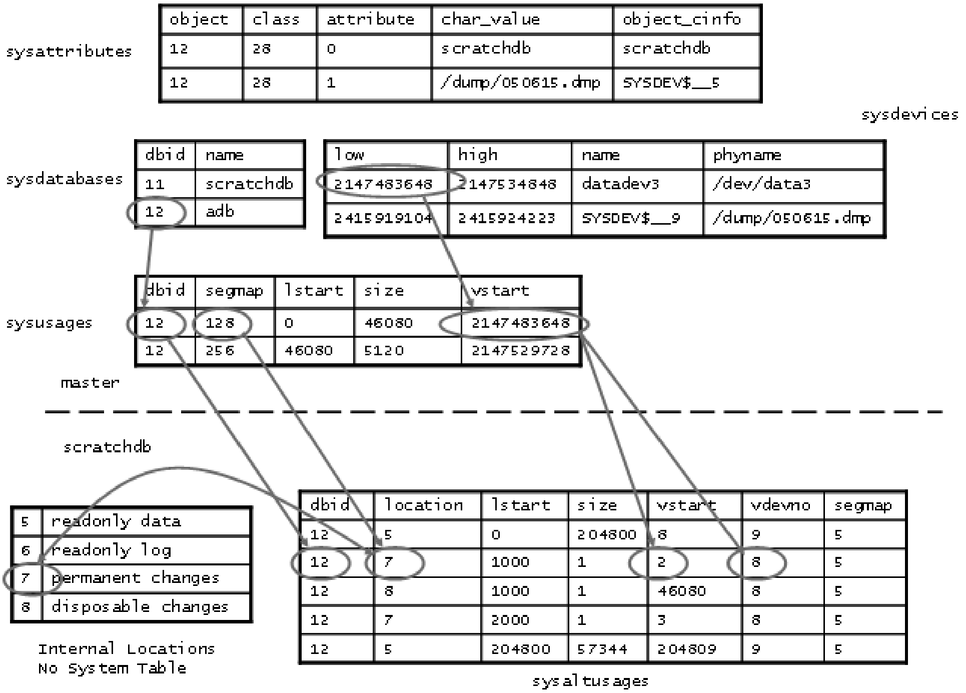
  A value of 5 and 6 means the location is in the database dump or its stripes, and a value of 7 or 8 means that the location is in the modified pages section. A value of 4 is used to fill the gaps for pages that are not physically available.
- lstart – is the logical page number of the start of the block of physically contiguous pages.

- size – is the number of logical pages in the block of physically contiguous pages.
- vstart – is the offset of the start of the contiguous block of pages on the device given by vdevno.
- vdevno – is the device number on which the contiguous block of pages resides.
- segmap – is a map of the segments to which this block of pages is allocated.

The sysaltusages table looks similar to Figure 3-2.

Note Because sysaltusages is a row-locked catalog, you may need to periodically use reorg to reclaim logically deleted space.

Figure 3-2: The sysaltusages table



sysattributes

object	class	attribute	char_value	object_cinfo
12	28	0	scratchdb	scratchdb
12	28	1	/dump/050615.dmp	SYSDEV$__5

sysdatabases

dbid	name
11	scratchdb
12	adb

sysdevices

low	high	name	phyname
2147483648	2147534848	datadev3	/dev/data3
2415919104	2415924223	SYSDEV$__9	/dump/050615.dmp

sysusages

dbid	segmap	lstart	size	vstart
12	128	0	46080	2147483648
12	256	46080	5120	2147529728

master

scratchdb

5	readonly data
6	readonly log
7	permanent changes
8	disposable changes

Internal Locations
No System Table

dbid	location	lstart	size	vstart	vdevno	segmap
12	5	0	204800	8	9	5
12	7	1000	1	2	8	5
12	8	1000	1	46080	8	5
12	7	2000	1	3	8	5
12	5	204800	57344	204809	9	5

sysaltusages

The scratch database

The scratch database stores the new sysaltusages table. The scratch database is used to provide flexibility as to where the sysaltusages table is located.

The scratch database can be any database (with some exceptions like master and temporary databases.) Sybase recommends that you dedicate a database that is used only as a scratch database, because:

- The size of sysaltusages may vary depending on the number of archive databases it supports. You cannot decrease the size of a database, but if it is too large, you can drop it and re-create a smaller database when required.
- It allows you to turn on the "trunc log on checkpoint" option so that the database log be automatically truncated.

Apart from hosting the sysaltusages table, this database is like any other. You can use threshold procedures and other space management mechanisms to manage space within the database.

You can specify a database that may be used as a scratch database by entering:

```
sp_dboption <db name>, "scratch database", "true"
```

Each archive database can be assigned to only one scratch database at a time, but multiple archive databases can use the same scratch database. If you have a large number of archive databases, you may want to define multiple scratch databases.

Working with an archive database

Many traditional database operations are possible on an archive database. However, user-defined transactions and commands that modify the database such as such as insert, update, and delete are not allowed.

A populated archive database is similar to a read-only database where the 'readonly' option has been applied using the sp_dboption stored procedure.

Configuring an archive database

This section provides information about how to create and configure an archive database.

Creating an archive database

You can create an archive database by issuing the create archive database command. The syntax is:

```
create archive database <db name>
    [on <db device> [= <size>]
        [, <db device> [= <size>] ] ... ]
    with scratch_database = <db name>
```

where:

- on – specifies the modified pages section. Adaptive Server requires traditional database storage to store modified pages. Use the on clause to specify the location and size of the modified pages section.
    - *db device* – specifies the database device on which you want to create your modified pages section.
    - *size* – specifies the size of the modified pages section you want to create. If *size* is omitted, 5120 pages are allocated.
- with scratch_database – (required) specifies the name of an existing database in which information about the archive database is maintained. The sysaltusages system table, which maps every logical page in the archive database onto a physical page, is stored in the scratch database.

Sizing the modified pages section

The modified pages section is used to store database pages that are modified or newly allocated.

- A page can be remapped to the modified pages section only once.
- Recovery is responsible for most page remappings.
- dbcc checkalloc also requires significant space allocation.
- The size of the modified pages section can be increased using the alter database command. However, to decrease the size of the modified pages section, you must drop the archive database and re-create it.

The minimum size of the modified pages section depends on how many pages are modified or newly allocated in the database. Many of these pages are modified by redo recovery and undo recovery.

Use the load database with norecovery command to minimize the number of modified pages and therefore the amount of space required in the modified pages section. There are downsides to doing this. For more information, see “Using load database with norecovery” on page 66.

Note dbcc checkalloc consumes a large amount of space in the modified pages section, even if you use the nofix option. When you run dbcc checkalloc, every allocation page (every 256th page) has information written to it. These allocation-page modifications are stored in the modified pages section, and mean that when you are using dbcc checkalloc, you need modified pages section that is at least 1/256th the size of the original database.

If you do not have enough space in the modified pages section, the command that requires the space is suspended and you see an error similar to:

```
There is no more space in the modified pages section for
the archive database <database name>. Use the ALTER
DATABASE command to increase the amount of space
available to the database.
```

To increase space in the modified pages section, either:

- Use alter database to increase the size of the modified pages section, or
- If you do not want to allocate additional space to the modified pages section, enter Ctrl+C to abort the current command.

Note You cannot use thresholds to manage the space in the modified pages section.

Increasing amount of space allocated to the modified pages section

You can use alter database to add space to the modified pages section of the archive database. Increasing the space in the modified pages section allows a suspended command to resume operation.

You can use alter database at any time to increase the size of the modified pages section, not only when space runs out.

```
alter database <db name>
    [ on <db device> [= <size>]
        [, <db device> {= <size>] ] ...]
```

Materializing an archive database

An archive database is a placeholder that is useful only once it has been loaded with a database dump. The load process does not actually copy pages, however, it materializes the database using page mapping.

Use the load database command to materialize an archive database. The syntax is:

```
load database <db name>
    from <dump device>
    [ [stripe on <stripe device>] ... ]
    [with [norecovery,][passwd=<password>]
```

where:

- *db name* – specifies the name of the archive database into which you want to load.
- *dump device* – specifies the name of the disk database dump from which you want to load the dump.
- *stripe device* – specifies additional disk database dump stripes.
- norecovery – indicates that the load database command will not run recovery, and that the database is brought online automatically after the load database command has completed. For more information, see “Using load database with norecovery” on page 66.

  Note norecovery has been introduced in Adaptive Server version 12.5.4 for archive database access. You cannot use norecovery on a traditional database.

- passwd =<*password*> – must be specified if the dump from which you are loading the archive database is password-protected. The password must be between 6 and 30 characters long.

Note You do not need to have Backup Server running when loading a dump into an archive database.

Using *load database with norecovery*

The with norecovery option of the load database command allows a database dump to be loaded into an archive database without recovering anything, reducing the time required to load. Many database pages can be modified or allocated during recovery, causing them to be stored in the modified pages section. Therefore, skipping recovery consumes minimum space in the modified pages section. The with norecovery option allows a quick view into an archive database.

If you use with norecovery, the database is brought online automatically.

However, using load database with norecovery for a database that requires recovery may leave it transactionally and physically inconsistent. Running dbcc checks on a physically inconsistent database may produce many errors.

Once you have loaded an archive database with norecovery, you must have sa_role or database owner privileges to use it.

Using logical devices with an archive database

You can use sp_addumpdevice to create a logical device from which an archive database can be loaded. The syntax is:

```
sp_addumpdevice ‘archive database’, ‘<logical name>’,
‘<physical name>’
```

After you have executed this command, you can now use the *logical name* instead of the *physical name* as the *dump device* or *stripe device* in a load database command.

load database limitations with an archive database

load database has the following limitations when used with an archive database:

- The database dump for an archive database is required to be a disk dump on a file system mounted on the local machine. This can be local storage or NFS storage. `load database ... at <remote server>` syntax is not supported, nor are database dumps on tape.
- Cross-architecture loads are not supported. The database dump and the load database command must be performed on the same architecture with respect to byte ordering.
- The database that was dumped must have the same page size as that used by the server that is hosting the archive database.

- The major version of the server on which the dump was taken must be earlier than or equal to the major version of the server hosting the archive database.
- The character set and sort order on the server on which the database dump was taken must be the same as the character set and sort order of the server hosting the archive database.
- The maximum size of any stripe in the database dump cannot exceed 32GB.

Note If the stripes are bigger than 32GB, specify additional stripes when you issue a dump database command. That way, the size of each stripe is reduced such that it may fall within the maximum size.

Bringing an archive database online

To bring an archive database online, use:

```
online database <db name>
```

where *db name* is the name of the archive database you want to bring online.

online database performs undo recovery during which modified and allocated pages may be remapped to the modified pages section.

You need not bring a database online if it has been loaded with norecovery, since the load automatically brings the database online without running the recovery undo pass.

Dropping an archive database

To drop an archive database, use:

```
drop database <db name>
```

where *db name* is the name of the database you want to drop.

When dropping an archive database, all the rows for that database are deleted from the sysaltusages table in the scratch database. This requires log space in the scratch database.

Note If you run out of log space in the scratch database while drop database is running, enter Ctrl+C, truncate the log in the scratch database, and retry the drop database command. Attempting to truncate the log in the scratch database *without* first entering Ctrl+C to end the drop database command results in the truncation being suspended as it waits on a lock that has been acquired by the drop command.

Using an archive database

This section provides information on the commands that can be run on an archive database.

Using SQL commands with an archive database

In addition to the commands already documented, (alter database, load database, online database, and drop database) the following SQL commands are allowed in an archive database:

- use
- select
- select into – where the target database is not an archive database.
- Cursor operations that perform reads, including:
    - declare cursor
    - deallocate cursor
    - open
    - fetch

  You cannot use an updateable cursor.
- checkpoint – is a supported command. However, the checkpoint process does not checkpoint an archive database automatically.

- execute is allowed as long as any statements that reference the archive database are allowed within the archive database. A transaction inside or outside a stored procedure is not permitted with an execute command.
- lock table
- readtext

Note DML commands including insert, update, and delete are not permitted, and you cannot start user transactions.

Using dbcc commands with an archive database

The following dbcc commands are allowed in an archive database:

- checkdb
- checkcatalog

  Note The fix version of checkcatalog is not supported.

- checktable
- checkindex
- checkalloc
- indexalloc
- tablealloc
- textalloc

dbcc commands prevent other users from accessing an archive database while they are executing. If you attempt to access an archive database while dbcc commands are being performed, you receive a message saying that the database is in single-user mode.

You can use variants of the above dbcc commands on an archive database that is online or offline. However, you can use a dbcc command with a fix option only on an archive database that is online.

Typical archive database command sequence

The following syntax could be a typical archive database command sequence.

First, create the scratch database if necessary, using the create database command.

```
create database scratchdb
      on datadev1 = 100
      log on logdev1 = 50
```

This creates a 150MB traditional database called scratchdb.

Use sp_dboption to designate the database you just created as a scratch database:

```
sp_dboption “scratchdb”, "scratch database", "true"
```

Create the archive database.

```
create archive database archivedb
      on datadev2 = 20
      with scratch_database = scratchdb
```

This creates an archive database called archivedb, with a 20MB modified pages section.

Materialize your archive database using load database:

```
load database archivedb
      from “/dev/dumps/050615/proddb_01.dmp”
      stripe on “/dev/dumps/050615/proddb_02.dmp”
```

Bring the database online:

```
online database archivedb
```

Check the consistency of the archive database using dbcc commands. For example:

```
dbcc checkdb(archivedb)
```

An object can be restored from the archive database using the select into or bcp commands. For example, to restore a table called orders from the archive database, use:

```
select * into proddb.dbo.orders from
      archivedb.dbo.orders
```

Security and permissions for an archive database

Permission to execute commands and stored procedures, and access to objects in an archive database is the same as for a traditional database loaded with the same database dump on the same server.

When an archive database is loaded with norecovery, access to that database is limited to users with sa_role, or the Database Owner.

Compressed dumps for an archive database

To use a compressed dump for an archive database, you must:

- Create the compressed dump with the with compression = <compression level> option of the dump database command.
- Create a memory pool for accessing the archive database.

Note Dumps generated with "compress::" cannot be loaded into an archive database. Therefore, any references to compression in this chapter refer to dumps generated using the with compression = <compression level> option.

Compression logic has changed in version 12.5.4. For information about compatibility issues, see "Compatibility issues for a compressed dump" on page 73.

Creating a compression memory pool

When Adaptive Server reads a page from a compressed dump, it selects a compressed block from the dump, decompresses it, and extracts the required page. The decompression in Adaptive Server is done using larger buffers from a special memory pool. The size of the pool is configured using:

```
sp_configure ‘compression memory size’, <size>
```

This is a dynamic configuration parameter, and the size is given in 2KB pages. If size is set to 0, no pool is created and a compressed dump cannot be loaded.

To determine the optimal size for your pool, consider these two factors:

- The block I/O used by the Backup Server. By default, this block I/O is 64KB, but it could have been changed using the with blocksize option in the dump database command.
- The number of *concurrent* users decompressing blocks within all archive databases. Each concurrent user requires two buffers each the same size as the block I/O.

As an absolute minimum, allow one concurrent user (two buffers) per archive database.

Migrating an archive database

sybmigrate does not migrate an archive database if an entire installation is being migrated.

sybmigrate only migrates an archive database 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 and downgrading an archive database

This section describes upgrade and downgrade instructions for an archive database.

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 online database command is executed.

If an Adaptive Server containing an archive database is upgraded, all the databases except the archive databases are upgraded. The archive database remains on the older version of Adaptive Server.

Sybase recommends you reload the archive database with a dump generated from an already upgraded database.

For more information about upgrading Adaptive Server, see the *Installation Guide* for your platform.

Downgrading an Adaptive Server with an archive database

When you are downgrading to a version of Adaptive Server that does not support archive databases, be aware of the following:

- If you must downgrade an Adaptive Server containing an archive database to a version of Adaptive Server that does not support archive databases, Sybase recommends you drop the archive database before you downgrade.

  To eliminate the new sysaltusages table, drop the scratch database before you perform the downgrade procedure. sysaltuages does not cause any problems if the scratch database is not dropped.

- Backup Server version 12.5.4 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.

Compatibility issues for a compressed dump

- You cannot load dumps generated with “compress::” into an archive database. There are no compatibility issues with dumps using this compression option on traditional databases.

- The format of a compressed dump generated with the with compression = <compression level> option has changed. Backup Server version 12.5.4 is the component that writes the new compression format. Therefore:

  - A compressed dump made using a 12.5.4 Backup Server can be loaded only into a 12.5.3 or earlier installation using a 12.5.4 Backup Server or higher.

- If you are using a 12.5.3 installation or earlier and want to use your dumps for an archive database, use a 12.5.4 version of the Backup Server to create compressed database dumps.

  Note A 12.5.4 Backup Server understands both 12.5.4 and earlier compression formats; therefore, you can use a 12.5.4 backup server for both dump and loads.

DDLGen support for archive database access

To generate DDL for all archive databases, use the extended filter option "OA."

```
ddlgen -Uroy -Proy123 -SHARBAR:1955 -TDB -N% -XOA
```

To generate DDL for a single archive database, use the syntax for normal databases. This example creates DDL for the archive database archivedb:

```
ddlgen -Uroy -Proy123 -SHARBAR:1955 -TDB -Narchivedb
```

Archive database limitations

An archive database has the following limitations:

- An archive database is read-only.
- An archive database is automatically in single-user mode when any command is run that results in changes to the archive database, such as dbcc commands.
- An archive database uses only database dumps on disk. Tape dumps are not supported.
- The database dump must be visible from the server that is hosting the archive database. Remote dumps are not supported.
- load tran is not supported in Adaptive Server version 12.5.4.
- dbcc checkstorage is not supported in Adaptive Server version 12.5.4.

- For an archive database to access compressed dumps, the dump must have been created with the with compression option rather than the "compress::" option.
- The checkpoint process does not automatically checkpoint an archive database. Use the checkpoint command to checkpoint an archive database.
- You cannot use sp_dbrecovery_order to specify an archive database in the database recovery sequence. Archive databases are recovered last, in their *dbid* order.
- When pages are cached in an archive database, the cached pages stay in the memory pool with the same page size as the server. So on a 2K server, the pages are always cached in a 2K pool. On a 16K server, the pages are always cached in a 16K pool.
- You cannot bind an archive database nor any object within that database to a user-defined cache. Objects within an archive database default to the default data cache.
- disk resize does not work on any device used by an archive database and that maps to a database dump or a transaction log.
- disk refit does not rebuild the master database's sysusages entries from any devices that are used by an archive database. This applies both to dump devices and those used for the modified pages section. Existing sysusages entries for an archive database remain however.
- An archive database cannot be replicated.
- An archive database does not fail over on a high-availability server.
- Free-space thresholds cannot be established on an archive database.

CHAPTER 4 Shared Directory Changes

Adaptive Server version 12.5.4 has made the following changes to its shared directory structure.

Shared directory changes

Adaptive Server version 12.5.4 includes a number of changes to the shared directory structure. Table 1 shows the directory

Table 1: Shared directory structure changes for UNIX and Linux platforms

Component	Old location	New location
Shared directory	*$SYBASE/shared-1_0*	*$SYBASE/shared*
Sybase Central	*$SYBASE/sybcent41*	*$SYBASE/shared/sybcentral43*
JRE	*$SYBASE/shared-1_0/JRE-1_3*	*$SYBASE/shared/jre142*
Shared JAR file	*$SYBASE/shared-1_0/lib*	*$SYBASE/shared/lib*

Table 2: Shared directory structure changes for Microsoft Windows platforms

Component	Old location	New location
Shared directory	*%SYBASE%\shared-1_0*	*%SYBASE%\Shared*
Sybase Central	*%SYBASE%\sybcent41*	*%SYBASE%\Shared\Sybase Central 4.3*
JRE	*%SYBASE%\shared-1_0\JRE-1_3*	*%SYBASE%\Shared\Sun\jre142*
Shared JAR file	*%SYBASE%\shared-1_0\lib*	*%SYBASE%\Shared\lib*

Note All Adaptive Server components have been modified to use the new directory structure.

Adaptive Server contains the following Java applications that are affected by this change:

- Sybase Central
- ASE Plug-in

- DDLGen
- ASE Replicator
- Web Services Producer and Consumer
- SQL Debugger
- Migration Tool
- Job Scheduler

CHAPTER 5 Sybase Driver Support

Adaptive Server version 12.5.4 contains new ODBC and OLE DB drivers developed by Sybase. The third-party re-branded ODBC and OLE DB Driver Kits included with previous versions are no longer shipped with Adaptive Server. The discontinued drivers continue to be supported as per the schedule described below.

The retired ODBC Driver Kit was installed in *%SYBASE%\ODBC*, and registered with the ODBC Driver Manager as “Sybase ASE ODBC Driver”. The new ASE ODBC Driver by Sybase is installed in *%SYBASE%\DataAccess\ODBC,* and registered as “Adaptive Server Enterprise”. The version shipping with Adaptive Server version 12.5.4 is version 12.5.1.510.

The retired OLE DB Driver Kit was installed in *%SYBASE%\OLEDB,* and used the provider short name of “Sybase.ASEOLEDBProvider” and the long name of “Sybase ASE OLE DB Provider”. The new ASE OLE DB Provider by Sybase is installed in *%SYBASE%\DataAccess\OLEDB,* and uses provider short name “ASEOLEDB”. The version shipping with Adaptive Server 12.5.4 is version 12.5.1.510.

Ongoing support for the third-party drivers

Existing customers with valid support contracts who are already using the third-party ODBC Driver and OLE DB Provider can continue to receive support for these products throughout the life cycle of ASE 12.5.

Until July 30, 2007, the support program will continue as it is today. After July 30, 2007, Sybase will address only P1 (severity level 1) issues. New updates to these drivers will be delivered as a standalone downloads.

Please note that the third-party ODBC Driver and OLE DB Provider you have today will only support ASE features up to and including ASE 12.5.3. Any issues found with these drivers will only be addressed by Sybase if they can be reproduced in ASE 12.5.3. If you use the third-party products with any releases of ASE beyond ASE 12.5.3, Sybase may not be able to identify problems or support fixes.

Please see the Sybase Web site at http://www.sybase.com/detail?id=1040652 for the End of Life notification for these drivers.

Migrating to new drivers

Sybase recommends that you start your migration process the new ODBC and OLE DB drivers as soon as possible. The third-party drivers will not support Adaptive Server features added in Adaptive Server version 12.5.4 and later.

See the *New Features for OpenServer 12.5.1 and SDK 12.5.1 for Windows, Linux and UNIX*, for instructions on migrating to the new drivers.

CHAPTER 6

Dynamically Loading TIBCO Libraries

Adaptive Server version 12.5.4 introduces support for dynamically loaded TIBCO JMS libraries.

Dynamically loading Adaptive Server messaging libraries

Adaptive Server version 12.5.4 dynamically loads the messaging libraries it needs to interact the with TIBCO EMS and IBM MQ message buses.

The Adaptive Server Messaging libraries contain the messaging logic and act as a wrapper on top of the messaging libraries provided by the vendors like TIBCO and IBM. These Adaptive Server messaging libraries are available with the purchase of RTDS 3.5 and later. After you install Adaptive Server, you must install RTDS 3.5 to install the Adaptive Server messaging DLLs. Install these libraries in the *$SYBASE/ASE-12_5/lib* directory.

In addition to Adaptive Server Messaging libraries, you must have the messaging libraries and DLLs from vendors (TIBCO and IBM.)

Once you have installed Adaptive Server Messaging libraries and the vendor specific message bus libraries, modify the LD_LIBRARY_PATH or the equivalent with the location of these libraries.

Adaptive Server Messaging libraries are located in *$SYBASE/ASE-12_5/lib*, by default.

You must also add the location of the vendor provided DLLs in the LD_LIBRARY_PATH or the equivalent. See

Adding TIBCO JMS DLL location information

Assuming TIBCO_HOME is the location where TIBCO EMS version 4.2 or 4.3 have been installed, the platform specific paths are as follows:

- On Solaris 32-bit platforms, add *$TIBCO_HOME/clients/c/lib* to LD_LIBRARY_PATH
- On Solaris 64-bit platforms add to following to LD_LIBRARY_PATH:
  - *$TIBCO_HOME/clients/c/lib/64*
  - *$TIBCO_HOME/clients/c/lib*
- On Linux 32-bit platforms add *$TIBCO_HOME/clients/c/lib* to LD_LIBRARY_PATH.
- On Windows 32-bit platforms add *%TIBCO_HOME%\clients\c\bin* to PATH.
- On HP-UX 64-bit platforms add the following to LD_LIBRARY_PATH:
  - *$TIBCO_HOME/clients/c/lib/64*
  - *$TIBCO_HOME/clients/c/lib*
- On IBM AIX 64-bit platforms add the following to LIBPATH:
  - *$TIBCO_HOME/clients/c/lib/64*
  - *$TIBCO_HOME/clients/c/lib*

Adding IBM MQ DLLs to LD_LIBRARY_PATH

Assuming MQM_HOME is where the MQ client library is installed, the platform specific paths are as follows:

- On Solaris 32-bit platforms, add *$MQM_HOME/lib* to LD_LIBRARY_PATH
- On Solaris 64-bit platforms, add *$MQM_HOME/lib64* to LD_LIBRARY_PATH
- On Linux 32-bit platforms add *$MQM_HOME/lib* to LD_LIBRARY_PATH.
- On Windows 32-bit platforms add *%MQM_HOME%\bin* to PATH.
- On HP-UX 64-bit platforms *$MQM_HOME/lib64* to LD_LIBRARY_PATH.

- On IBM AIX 64-bit platforms add *$MQM_HOME/lib64* to LIBPATH.

Note MQM_HOME is */opt/mqm* for Solaris, Linux and HP-UX and */usr/mqm* on IBM AIX. On Windows it is the directory where Websphere MQ is installed.

Additionally, sp_configure 'enable real time messaging' has been modified to accept the type of messaging that you want to enable:

- `sp_configure 'enable real time messaging', 1` enables both TIBCO JMS and IBM MQ messaging. This command succeeds if Adaptive Server can locate the DLL libraries for TIBCO JMS and IBM MQ.
- `sp_configure 'enable real time messaging', 1, 'tibco_jms'` enables TIBCO JMS messaging only. This command succeeds if Adaptive Server can locate the DLL libraries for TIBCO JMS.
- `sp_configure 'enable real time messaging', 1, 'ibm_mq'` enables IBM MQ messaging only. This command succeeds if Adaptive Server can locate the DLL libraries for IBM MQ

CHAPTER 7 JRE Support

JRE support in Adaptive Server 12.5.4

Adaptive Server version 12.5.4 includes JRE 1.4. JRE 1.4 is installed in full and typical installations by default, and in custom installations whenever a component that requires the JRE to run is selected for installation.

CHAPTER 8 Adaptive Server Plug-in Changes

Adaptive Server plug-in support

The Sybase Central Adaptive Server plug-in supports archive database access and configuring password complexity options in Adaptive Server version 12.5.4.

Use the Sybase Central Adaptive Server plug-in to manage archive databases. There is a new folder called Archive Databases under the Databases folder.

Use the Sybase Central Adaptive Server plug-in to configure Adaptive Server for password complexity options from the Server Property Sheet. For Adaptive Server version 12.5.4, there is a new tab in the Server Property Sheet called Login Password Configuration.

For information on using the Adaptive Server plug-in to manage archive database access, see Chapter 19, “Controlling Access to Adaptive Server” in *Managing Adaptive Server Enterprise*. For information on using the Adapitve Server plug-in to configure password complexity options, see Chapter 7, “Managing Databases” in *Managing Adaptive Server Enterpise*.

CHAPTER 9 Changes to Monitoring Tables

Adaptive Server 12.5.4 makes the following change to monitoring tables.

Changed monitoring tables

Table 9-1 describes updates to monitoring tables.

Table 9-1: Updates to monitoring tables

Column name	Monitoring table	Description
RowsAffected	monSysStatement andmonProcessStatement	Indicates the number of rows affected by the current statement. Helpful when looking for queries that may be using an inefficient query plan, because these queries probably show a high number of logical I/Os per returned row.

CHAPTER 10 Shared Memory in Windows Terminal Server Environments

Adaptive Server shared memory when running on Windows Terminal Server environments

To accommodate Windows Terminal Server shared memory requirements, Adaptive Server version 12.5.4 introduces the new environment variable SYBASE_TS_MODE.

Windows Terminal Server is a feature of Microsoft Windows that allows multiple remote users to log into the Windows system at once. Each user connects from a local workstation to a central Windows server, and is then given a virtual windows environment which appears back on the local workstation. The Windows server keeps the memory spaces and other resource allocations of these terminal server sessions entirely separate, so that users are largely unaware of the existence of other terminal server sessions. In addition there can be a user logged in to the machine in the normal way, this is referred to as the 'console session'.

The Adaptive Server data server can be started in a terminal server session, or it may start as a windows service in which case it is regarded as part of the console session. Adaptive Server shared memory regions are normally not accessible from other terminal server sessions, and thus certain tools such as performance monitors, and Sybase's internal diagnostic tool (sybmon) will not work unless started from the same session that owns the Adaptive Server, as they need to connect to the shared memory region of that server. If Adaptive Server is running as a service these tools only work if started from the console session.

Adaptive Server 12.5.4 looks for an environment variable SYBASE_TS_MODE, and if this is set to GLOBAL then Adaptive Server creates its shared memory so that it is available across all terminal server sessions. Similarly, changes made to the performance monitor and sybmon also look for this environment variable and connect to the shared memory globally if it is set to GLOBAL. In this way, you can configure a remote terminal session to monitor Adaptive Server running on the console session or on another terminal server session. SYBASE_TS_MODE should normally be created as a system environment variable and the Windows machine restarted to bring it into effect.

SYBASE_TS_MODE is not supported by Windows NT 4.0 Workstation and Server, and Windows 2000 Professional.

The default behavior of this variable if it is not set, is to create shared memory under the local terminal server session only.

Warning! SYBASE_TS_MODE must not be set on a machine that does not support terminal services, as this causes shared memory creation to fail and Adaptive Server will not start.

CHAPTER 11 Adaptive Server Enhancements on Linux Platforms

This chapter contains information about Adaptive Server enhancement on Linux platforms.

Large memory support enhancement

Large memory support allows you to set the extended cache size configuration parameter to a value other than 0.

In previous versions of Adaptive Server, the wash size of the smallest I/O memory pool in the primary data caches was set to a fixed value. Any configured value was overridden.

Adapvie Server version 12.5.4 allows you to configure the wash size for for the smallest I/O memory pool in the primary data cache.

DIRECTIO support

Adaptive Server version 12.5.4 does not does not support DIRECTIO on file system devices.

Large Memory Support and POSIX Async I/O

Adaptive Server version 12.5.4 supports the large memory support feature on Linux platforms configured to use POSIX Async I/O.

CHAPTER 12

Changes to Stored Procedures, Functions, and Commands

This chapter contains major changes in Adaptive Server version 12.5.4 not documented elsewhere.

New syntax for *shutdown*

Adaptive Server version 12.5.4 includes the following new syntax for shutdown:

```
shutdown [srvname] [with {wait [="hh:mm:ss"] | nowait}]]
```

Parameters

srvname

is the logical name by which the Backup Server is known in the Adaptive Server sysservers system table. This parameter is not required when shutting down the local Adaptive Server.

with wait

is the default. This shuts down the Adaptive Server or Backup Server gracefully.

hh:mm:ss

is an optional setting that specifies the maximum time the server waits for all running or sleeping processes to finish their job.

with nowait

shuts down the Adaptive Server or Backup Server immediately, without waiting for currently executing statements to finish.

Note Using shutdown with nowait can lead to gaps in IDENTITY column values.

Specifying a wait time

When the server prepares to shut down, it:

1. Performs checkpoint on all the databases.
2. Prevents any new user from logging in.
3. Waits for all running or sleeping processes to finish their job.
4. Performs another checkpoint on the databases, this time with a flag indicating it needs to flush:
    - All the dynamic thresholds in mixed log-data databases.
    - All the object statistics.
    - The values of the identity fields to avoid holes after recovery.

When you use with wait with the *hh:mm:ss option*, the time you specify is not the maximum total time Adaptive Server takes to shut itself down. Instead, Adaptive Server takes into account the time it takes to perform the first checkpoint, and automatically subtracts this from the time you specified.

For example, if you specify a maximum wait time of 20 minutes and the first checkpoint takes 3 minutes, Adaptive Server allows up to 17 minutes for the processes to finish. If for some reason the second checkpoint takes longer, however, this is not calculated into the with wait *hh:mm:ss* parameter you specify.

Adaptive Server also allows a checkpoint to take longer than the time you specify in with wait *hh:mm:ss*. For example, if you specify a wait time of 10 minutes but the first checkpoint takes 20 minutes to complete, Adaptive Server does not interrupt checkpoint midway, but instead waits for checkpoint to complete. When this occurs, Adaptive Server immediately begins to shut down after checkpoint is complete, since the time you specified has passed, and runs the last checkpoint with the flag informing you of the flushes you must perform.

Expanded *select ** syntax

When the source text of a stored procedure or trigger is stored in the system table syscomments a query using select * is stored in syscomments expanding the column-list referenced in the select *.

For example a select * from a table containing the columns col1 and col2 is stored as:

```
select <table>.col1, <table>.col2 from <table>
```

In 12.5.4. the expanding of the column-list is enhanced in a way that identifiers (table-names, column-names and so on) are checked if they comply with the rules for identifiers.

For example, we have a table with the columns col1 and 2col. The second column-name starts with a number which can only be done using brackets in the create table statement. These are the bracketed identifiers.

When performing a select * in a stored procedure or trigger from that table, the text in syscomments looks similar to:

```
select <table>.col1, <table>[2col] from <table>
```

For all identifiers used in the text expanding a select *, brackets are added when the identifier does not comply with the rules for identifiers.

You must add brackets around identifiers to make sure Adaptive Server can use the SQL-text while performing an upgrade to a more recent release.

dump database and *load database* with verification

In Adaptive Server 12.5.4, the dump database and load database commands introduce an option to allow you to dump and load databases with verification. The syntax for dump database is:

```
dump database <dbname> with verify [ = header | full ]
```

The syntax for load database is:

```
load database <dbname> with verify[only] [= header |
full ]
```

When you execute dump database, Backup Server performs minimal header and structural row checks for data pages as they are being copied to the archives. There are no structural checks done at this time to GAM, OAM, allocation pages, indexes, text, or log pages.

You can perform the same checks when loading an archive using the command load with verify[only].

An archive can be checked without a physical load, using the load database with verifyonly command: this option displays the dump header information similar to the load option of with headeronly.

Allowing updates to system catalogs

The server-wide configuration option allow updates to system catalogs takes precedence over the stored procedure settings for allow updates. If not enabled server wide, stored procedure settings determine whether you can modify system catalogs.

Modulo arithmetic for numeric datatypes

In Adaptive Server version 12.5.4, you can perform modulo arithmetic on reals, floats, decimals, numerics, as well as on integers.

New function to support IPv6 IP address requirements

IPv6 architecture an IP address length of 64 bytes. As the size of sysprocesses could not be increased, Adaptive Server needed a way to retrieve the full IP address. Adaptive Server version 12.5.4 introduces a new function to return information from the pss:

```
pssinfo(<spid | 0>, '<pss field>')
```

where:

- *spid* – Process ID. When you enter 0, the current process is used.

- *pss field* – valid values are:
  - ipaddr – Client IP Address
  - extusername – When using external authentication like (PAM, LDAP), it returns the external PAM or LDAP user name used.
  - dn – Distinguished Name when using LDAP authentication.

  The pssinfo function also includes the option to display the external user name and the distinguish name.

Functions to decode external transactions

Adaptive Server version 12.5.4 adds two functions to decode external transactions. A man page for each follows.

xa_bqual

Description

Returns the binary version of the bqual component of an ASCII XA transaction ID.

Syntax

xa_bqual(*xid*, 0)

Parameters

xid

is the ID of an Adaptive Server transaction, obtained from the xactname column in systransactions or from sp_transactions.

0

is reserved for future use

Examples

Example 1 Returns “0x227f06ca80”, the binary translation of the branch qualifier for the Adaptive Server transaction ID “0000000A_IphIT596iC7bF2#AUfkzaM_8DY6OE0”. The Adaptive Server transaction ID is first obtained using sp_transactions:

```
1> sp_transactions
```

```
xactkey                        type     coordinator starttime           st
ate      connection dbid  spid  loid  failover    srvname  namelen  xactna
me
------------------------------ -------- ----------- ------------------  --
```

```
-------- ---------- ----- ----- ----- ---------- -------  -------  ------
--------------------------------
0x531600000600000017e4885b0700 External XA          Dec 9 2005  5:15PM In
Command  Attached     7   20    877  Resident Tx  NULL         39
0000000A_IphIT596iC7bF2#AUfkzaM_8DY6OE0
```

```
1> select xa_bqual("0000000A_IphIT596iC7bF2#AUfkzaM_8DY6OE0", 0)
2> go
```

```
...

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

0x227f06ca80
```

Example 2 xa_bqual is often used together with xa_gtrid. This example returns the global transaction IDs and branch qualifiers from all rows in systransactions where its coordinator column is the value of “3”:

```
1> select gtrid=xa_gtrid(xactname,0),
    bqual=xa_bqual(xactname,0)
    from systransactions where coordinator = 3
2> go
        gtrid

        bqual

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

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

         0xb1946cdc52464a61cba42fe4e0f5232b

         0x227f06ca80
```

Usage

If an external transaction is blocked on Adaptive Server and you are using sp_lock and sp_transactions to identify the blocking transaction, you can use the XA transaction manager to terminate the global transaction. However, when you execute sp_transactions, the value of *xactname* it returns is in ASCII string format, while XA Server uses an undecoded binary value. Using xa_bqual thus allows you to determine the bqual portion of the transaction name in a format that can be understood by the XA transaction manager.

xa_bqual returns:

- The translated version of this string that follows the second “_” (underscore) and precedes either the third “_” or end-of-string value, whichever comes first.

- NULL if the transaction ID cannot be decoded, or is in an unexpected format.

Note xa_bqual does not perform a validation check on the xid, but only returns a translated string.

Standards

ANSI SQL – Compliance level: Transact-SQL extension.

Permissions

Any user can use xa_bqual.

See also

Functions xa_gtrid

Stored procedures sp_lock, sp_transactions

xa_gtrid

Description

Returns the binary version of the gtrid component of an ASCII XA transaction ID.

Syntax

xa_gtrid(*xactname*, *int*)

Parameters

xid

is the ID of an Adaptive Server transaction, obtained from the xactname column in systransactions or from sp_transactions.

0

is reserved for future use

Examples

Example 1 In this typical situation, returns “0x227f06ca80,” the binary translation of the branch qualifier, and “0xb1946cdc52464a61cba42fe4e0f5232b,” the global transaction ID, for the Adaptive Server transaction ID “0000000A_IphIT596iC7bF2#AUfkzaM_8DY6OE0”:

```
1> select xa_gtrid("0000000A_IphIT596iC7bF2#AUfkzaM_8DY6OE0", 0)
2> go

          ...

------------------------------------------------------------------
         0xb1946cdc52464a61cba42fe4e0f5232b

(1 row affected)
```

Example 2 xa_bqual is often used together with xa_gtrid. This example returns the global transaction IDs and branch qualifiers from all rows in systransactions where its coordinator column is the value of “3”:

```
1> select gtrid=xa_gtrid(xactname,0),
    bqual=xa_bqual(xactname,0)
    from systransactions where coordinator = 3
2> go
        gtrid

        bqual

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

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

         0xb1946cdc52464a61cba42fe4e0f5232b

         0x227f06ca80
```

Usage

If an external transaction is blocked on Adaptive Server and you are using sp_lock and sp_transactions to identify the blocking transaction, you can use the XA transaction manager to terminate the global transaction. However, when you execute sp_transactions, the value of *xactname* it returns is in ASCII string format, while XA Server uses an undecoded binary value. Using xa_gtrid thus allows you to determine the gtrid portion of the transaction name in a format that can be understood by the XA transaction manager.

xa_gtrid returns:

- The translation version of tis string that follows the first “_” (underscore) and preceeds either the second “_” or end-of-string value, whichever comes first.
- NULL if the transaction ID cannot be decoded, or is in an unexpected format.

Note xa_gtrid does not perform a validation check on the xid, but only returns a translated string.

Standards

ANSI SQL – Compliance level: Transact-SQL extension.

Permissions

Any user can use xa_gtrid.

See also

Functions xa_bqual

Stored procedures sp_lock, sp_transactions

Index