SYBASE®
An SAP® Company

New Features Guide

Adaptive Server® Enterprise 15.7 SP121

DOCUMENT ID: DC00641-01-1570121-01
LAST REVISED: April 2014

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

Contents

Certicom Replacement

Certicom software, which provides cryptography services for securing storage and transmission of sensitive information, is no longer supported by SAP® Sybase® products. These services have been replaced by alternate providers, as indicated in the documentation for each SAP Sybase product.

OpenSSL is the Adaptive Server® supported provider on all platforms. For information about OpenSSL, see the OpenSSL Web site at *http://www.openSSL.org"*. Changes due to the provider replacement are:

Certificate Management

These certificate utilities are no longer supported:

- `certreq`
- `certauth`
- `certpk12`

As a replacement, Adaptive Server includes the `openssl` open source utility in:

- (UNIX) `$SYBASE/$SYBASE_OCS/bin/openssl`
- (Windows) `%SYBASE%\%SYBASE_OCS%\bin\openssl`

Use `openssl` to accomplish all certificate management tasks previously implemented by `certreq`, `certauth`, and `certpk12`. For information about the use of the `openssl` utility, go to *http://www.openssl.org/docs/apps/openssl.html*.

Enabling FIPS Encryption

In previous releases, enabling the **FIPS login password encryption** parameter specified the use of FIPS 140-2 compliant cryptographic module for the encryption of passwords in transmission, in memory, and on disk. For Adaptive Server 15.7 SP60, enabling this parameter specifies that the FIPS 140-2 compliant cryptographic module is used for all encryption related operations.

Also in previous releases, FIPS Encryption was turned on by default. However, for Adaptive Server 15.7 SP60 it must be explicitly enabled. Client libraries must also enable FIPS to complete FIPS configuration.

Note: FIPS Certification is not supported for IBM AIX and Linux on POWER platforms.

Certificate Generation

OpenSSL in FIPS mode is strictly controlled by OpenSSL security. This means that some certificates that worked with the Certicom FIPS module may no longer work using OpenSSL. In particular, the use of MD5 algorithm is not FIPS 140-2 compliant. Old certificates using

this algorithm must be replaced in order to enable the **FIPS login password encryption** parameter.

When generating a FIPS compliant certificate, FIPS 140-2 compliant algorithms must be used. Private keys must be in pkcs8 format and encrypted with an OpenSSL FIPS 140-2 compliant algorithm. To determine what algorithm is used to encrypt a private key, enter the following command:

```
openssl asn1parse -in <Encrypted Key Filename> -inform PEM
```

To convert the key to the proper format use the following command:

```
openssl pkcs8 -in <Non-FIPS compliant Encrypted Key Filename>
-topk8 -out <FIPS compliant Encrypted Key Filename>
-v1 PBE-SHA1-3DES
```

Digital Signature RSA Encryption Algorithms

If RSA encryption algorithms are used for the digital signature, the RSA key size must be at least 1024 bit.

Cipher Support

When configuring FIPS cipher suites, the supported cipher suites have changed. These are the supported cipher suites for FIPS:

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

Reducing load database Time

Use the **optimize dump for faster load** configuration parameter to optimize **dump database** for a faster **load database** time.

The time it takes to perform a **load database** (either **full** or **cumulative**), is dependent on two primary factors:

- The time it takes for Backup Server to physically copy the pages from the dump archive into the database.
- The time it takes for **load database** to recover the loaded database pages. Recovery time is largely dictated by the amount of work that needs to be redone.

By reducing the amount of work that **load database** recovery has to redo, the time it takes to perform a **load database** decreases. The amount of recovery work that needs to be redone is related to the volume of transactional work that is on-going while **dump database** (**full** or **cumulative**) copies database pages using Backup Server, from the database into the dump archive. In versions earlier than 15.7 SP121, a single iteration in which all the pages that must be copied by Backup Server to the archive is done by **dump database**.

As of 15.7 SP121, multiple iterations are supported. That is, at the end of the first iteration, changes that have been made to the database while the first iteration was active, are copied by Backup Server to the archive in a second iteration. The set of changes in each iteration is less than those in the previous iteration, giving **load database** less changes to recover. A maximum of four iterations can be performed.

The **optimize dump for faster load** configuration parameter is used to configure this behavior. The syntax is:

```
sp_configure "optimize dump for faster load", <percentage>
```

When `<percentage>` is set to 0, a single iteration is done. This is the same behavior as in versions earlier than 15.7 SP121.

When `<percentage>` is set to a non-zero value, it represents the minimum percentage of data pages (not log pages) in the database that have changed relative to the total number of in-use data pages in the database since the start of the current iteration in order for the next iteration to be performed (up to a maximum of 4 iterations).

For example, when `<percentage>` is set to a value of 2, and if the percentage of changed data pages in the database since the start of the first iteration is greater than 2% of the total in-use data pages in the database, then at the end of the first iteration of copying pages, another iteration of copying pages is performed. In this case, the second iteration of copying pages starts, and a similar test as to whether to start a third iteration is performed at the end of the second iteration.

When optimizing a dump for faster load, it will take longer to perform a **dump database**.

Cyclic Redundancy Checks for dump database

A cyclic redundancy check has been added for accidental changes to raw data for database or transaction dumps created with compression. Use the cyclic redundancy check for checking and verifying that the compression blocks can be correctly read and decompressed.

The syntax is:

```
dump database database_name to dump_device with
compression=n,verify={crc | read_after_write}
load database database_name from dump_device with verify[only]=crc
```

Where:

- **verify=crc** – indicates that you are performing a cyclic redundancy check.
- **verify=read_after_write** – Backup Server rereads every compressed block after writing and decompresses it. If Backup Server finds an error, it prints the offset in the file in which it finds the error. **verify=read_after_write** is only applicable with the **dump database** command.

This example verifies database `new_dump` before dumping it to the `mydumpdev` device:

```
dump database new_dump to mydumpdev with
compression=x,verify=read_after_write
```

This example performs a cyclic redundancy check as it loads the `new_dump` database dump:

```
load database new_dump from mydumpdev with verify[only]=crc
```

This example performs a cyclic redundancy check and rereads every compressed block before dumping database `new_dump` to the `mydumpdev` device:

```
dump database new_dump to mydumpdev with
compression=x,verify=read_after_write,verify=crc
```

Usage:

- Dumps created without the **verify=crc** parameter use the same format as earlier versions of Backup Server.
- The **verify=crc** option is ignored if the database was not originally dumped using **verify=crc**.
- You cannot load dumps that include cyclic redundancy checks with versions of Backup Server that do not include this functionality.
- **verify={crc | read_after_write}** checks are applicable only for files created using the **with compression** parameter.

- **verify=crc** works with any file type, including raw devices, disk files, tapes, pipes, or APIs. However, **verify=read_after_write** requires a 'seek back' for rereading the block, and is applicable only with raw devices and disk files.
- If you include **verify={crc | read_after_write}** parameters that are not applicable, they are ignored and no error message is raised.

SAP JVM Support

SAP JRE is used to support Java applications.

By default, the SAP JRE is installed in:

```
$SYBASE/shared/SAPJRE-7_*
```

The installer automatically sets the *SAP_JRE7*, *SAP_JRE7_32*, and *SAP_JRE7_64* environment variables.

Note: (IBM AIX only) You must set the data size resource limit to `unlimited` when using any Java application:

```
limit datasize unlimited
```

See your operating system documentation.

Shrink Database Backlink Behavior Change

Improve performance for backlinking pointers for shrink database operations.

The shrink database operations for 15.7 SP100 and later, require all text and image columns have backlinking pointers to the home data row in the first text page. By default, the text and backlinking pointers are created and maintained if the table contains text, image, or unitext columns. However, this can impact the performance of these commands:

- **create clustered index**
- **reorg rebuild**
- **reorg defrag**
- **alter table** (which involves data copy)
  - unpartition
  - split or move partition
  - drop a column
  - add a NOT NULL column
  - change table scheme from APL to DOL, or vice-versa
  - modify any of these properties of a column
    - the datatype
    - from NULL to NOT NULL, or vice-versa
    - decrease length
    - increase the length of a number column (for example, from tinyint to int)

To improve performance, 15.7 SP121 and later maintains backlinking pointers for the above commands when tables are set up for replication using text and image backlinking pointers that use, **sp_reptostandby**, **sp_setreptable**, or **sp_setrepcol** without using the **use_index** parameter. Otherwise, the tables are marked without text and image backlinking pointers after the command execution.

Important

Since shrink database operations require text and image backlinking pointers be maintained, SAP strongly suggests you follow these steps before shrinking a database:

- Run the proposed shrink database command, adding the qualifier **with check_only**
- Run **dbcc shrinkdb_setup** for each table noted by that command as having its text affected

If you have already initiated a shrink database operation, do not run any of above commands in the database being shrunk, otherwise, the operation can be halted with error 5066:

```
Table <table name> contains off-row columns that do not link back to
their owning rows.
```

Enhanced Checking Functionality for Shrinking Databases

Additional checking capabilities for the shrink database functionality have been added.

The alter database command is used to shrink databases. The syntax is:

```
alter database database_name
. . .
   off database_device {=size | [from page_number] [to page_number]}
    [, database_device…]
    [with timeout='time']
    [with check_only]
```

The **with check_only** command option is used to check the expected results of shrinking the database rather than actually shrinking the database.

In 15.7 SP100, the **with check_only** option checks for:

- tables with text columns that are not marked as having text back-linked to their owning rows.

  All text and image columns are required to have back-linked pointers to the home data row in the first text page.
- sufficient room in the remainder of the database to accommodate everything that must be moved.

  Any indication of a potential problem is reported.

In 15.7 SP121, the **with check_only** option additionally checks and reports any potential problems where a significant amount of time could be spent sorting the index.

When checking the indexes, if the results indicate that there could be enough duplicate key entries that the command will spend a significant amount of time sorting the index, the index is reported as a problem. The recommendation is that the index should be dropped before shrinking the database and the index be re-created after the database is shrunk.

Results of the checks being done by **with check_only** can be compromised by any other work being performed on in the database while the checks are running, or by work that was done after the checks are run but before the actual shrink is completed.

sp_monitorconfig Permission Changes

The permission requirements for **sp_monitorconfig** have changed.

To execute **sp_monitorconfig**:

- With granular permissions enabled, you must be a user with mon_role or have `manage server` privilege.
- With granular permissions disabled, you must be a user with either mon_role or sa_role.

Changes for alter table add | drop partition

The **enable utility lvl 0 scan wait** configuration parameter allows you to run **alter table ... add | drop partition** commands while the server runs isolation level 0 scans.

enable utility lvl 0 scan wait

Default value	0
Range of values	0 (off), 1(on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator
Configuration group	Application Functionality

The default value for **enable utility lvl 0 scan wait** depends on the value to which **enable functionality group** is set. If you set enable functionality group to:

- 0 – the default value for **enable utility lvl 0 scan wait** is 0
- 1 – the default value for **enable utility lvl 0 scan wait** is 1

When setting **enable utility lvl 0 scan wait** to 1, it uses a value of 1 regardless of what you set **enable functionality group** to.

Option to Control Wait Behavior for Uncommitted Inserts

The configuration option **wait on uncommitted insert** allows you to control the wait behavior of the **update**,**insert**, and **delete** commands for an uncommitted insert.

wait on uncommitted insert

Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator
Configuration group	SQL Server Administration

wait on uncommitted insert can only be used for DOL tables on transaction isolation 0, 1, and 2, and does not affect the behavior of transaction isolation 3.

When **wait on uncommitted insert** is set to 1:

- at isolation level 0, delete and update queries block on uncommitted inserted rows with the key value of interest, and select queries read dirty and do not block on uncommitted inserts
- at isolation levels 1 and 2, select, delete, and update queries, all block on uncommitted inserted rows

When **wait on uncommitted insert** is set to 0, the behavior of select, update, delete, and insert is the same as in previous releases.

Note: When **wait on uncommitted insert** is set to 1 (to wait for uncommitted inserted rows) concurrency might descend and deadlocks can be encountered where they were not encountered before.

Support for Remote Dump Host Control

The remote access control feature introduced in 15.7 SP50 prevents remote dumps and loads, and execution of remote procedure calls (RPCs) from any client or server that is running on unauthorized servers.

To support remote host control, **allow_hosts_list** has been added to the `svrbuild.backup.server.rs` resource file for configuring Backup Server using **srvbuild** and **srvbuildres**. This attribute accepts a comma separated list of hosts allowed.

Support for Parallel Index Creation on Partitioned Table with Empty Partitions

The **create index** command has been enhanced to support the creation of non-clustered local index in parallel for a partitioned table when some of the partitions are empty.

Dump Commands Support for Mirrored Devices

The **dump database** and **dump transaction** commands now take into account when a device is mirrored and will store the path to mirror devices along with regular devices in the dump header.

In previous releases, only information about primary devices was stored in the dump header file.

The **sybdumptran** utility will first attempt to open the primary device, and in the event the primary device cannot be opened, the mirrored device is opened, if one exists.

Extended Maximum Variable Length of a Datatype

The full maximum datatype length is now supported when creating a user-defined datatype. In previous releases, length of a variable datatype was limited to server page size.

The global variable **@@maxvarlen** has been introduced to check the maximum possible variable length allowed when creating a user-defined datatype.

Diagnostic Enhancements for 'Proc Cache Header' Memory Pool

A new diagnostic tool for measuring 'Proc Cache Header' performance has been implemented.

Note: Use this tool only under the guidance of the SAP support team.

1. Enable the diagnostic using:
   ```
   dbcc mempool("Proc Cache Header","enable diagnostics")
   ```
2. To capture the diagnostic messages, enable the **rapidlog** configuration parameters:
   ```
   sp_configure 'enable rapidlog', 2
   ```
   ```
   sp_configure 'rapidlog buffer size', <size>
   ```
   ```
   sp_configure 'rapidlog max files', <X>
   ```
   The default size for **rapidlog buffer** size is 1024.
   The default size for **rapidlog max files** size is 99999999.
   The server does not need to be restarted.
3. You can manually flush the buffer to the file by issuing:
   ```
   dbcc rapidlog("!FLUSH")
   ```
4. Send the compressed `$SYBASE/rapidlog*` files to the SAP support team.
   The diagnostic output is saved to `rapidlog*` files under the `$SYBASE` directory.

To disable diagnostic use:

```
dbcc mempool("Proc Cache Header","disable diagnostics")
```

Configuration Parameters for Diagnostic Tool

Use the **rapidlog** configuration parameters to capture messages for the 'Proc Cache Header' memory pool diagnostic tool.

enable rapidlog

Default value	0
Range of values	0–255
Status	Dynamic
Display level	
Required role	System administrator

Configuration group	Error Log

rapidlog buffer size

Default value	1024
Range of values	1024–2000000
Status	Dynamic
Display level	
Required role	System administrator
Configuration group	Error Log

rapidlog max files

Default value	99999999
Range of values	1–99999999
Status	Dynamic
Display level	
Required role	System administrator
Configuration group	Error Log

Changes to System Tables

The `Description` and `Label` column widths for monitoring tables are extended.

Table	Changes
`monTables`	`Description` column width changed to 512 characters from 255 characters.
`monTableColumns`	• `Description` column width changed to 512 characters from 255 characters. • `Label` column width changed to 150 characters from 50 characters.

Known Issues

Known issues at the time of the 15.7 SP121 release.

Columns with Name sybdropcol <column_id> <object_id>

Issuing **split** or **move partition** on user tables that:

- Include any columns named `sybdropcol_<column_id>_<object_id>`
- Have undergone **alter table drop column** with no data copy

may fail, and issue error message 2705:

```
 Level 16, State 1: Line 1: Column names in each
table must be unique. Column name '<colname>' in table
'#<table name>' is specified more than once.
```

HP Asynchronous I/O Functionality

Support for the `aio_reap()` functionality has been added on the HP platform.

The HP AioEnh enhanced bundle provides the functionality for `aio_reap()`. To download the HP AiOEnh product, see the HP Web site at: *https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=AioEnh"*.

In addition to the AioEnh bundle, install the following patches:

- PHKL_42844
- PHCO_42837

Documentation Updates

Read about updates to the documentation released with Adaptive Server.

Upgrading Job Scheduler

The steps for upgrading Job Scheduler after an upgrade have changed for 15.7 SP121. After upgrading to a new server, upgrade Job Scheduler using these steps.

Note: The directory with the **isql** executable must be in your path:

- UNIX - `$SYBASE/$SYBASE_OCS/bin`
- Windows - `%SYBASE%\%SYBASE_OCS%\bin`

---

1. Copy the directory services entry for JSAGENT (or jsagent) from the old server to the new server.
2. Make sure the new server is running.
3. Ensure that at least 9000 locks are configured. If the number of locks on the server is fewer than 9000, increase it:

   ```
   1> sp_configure "number of locks", 9000
   2> go
   ```
4. Before you restart Adaptive Server and run Job Scheduler, run the **installjsdb** script to update Job Scheduler tables and stored procedures:
   a. Disable Job Scheduler:

      ```
      1> sp_configure "enable job scheduler", 0
      2> go
      1> sybmgmtdb..sp_sjobcontrol @name=NULL, @option="stop_js"
      2> go
      ```
   b. Run the **installjsdb** script:

      For UNIX:

      ```
      isql –Usa –Psa_password –Sservername
      –i$SYBASE/$SYBASE_ASE/scripts/installjsdb
      ```

      For Windows:

      ```
      isql –Usa –Psa_password –Sservername
      –i%SYBASE%\%SYBASE_ASE%\scripts\installjsdb
      ```

      The `installjsdb` script looks for the `sybmgmtdb` database.
   c. Enable Job Scheduler:

      ```
       sp_configure "enable job scheduler", 1
      ```
   d. To start Job Scheduler, either restart the server, or enter:

      ```
      1> use sybmgmtdb
      2> go
      ```

```
1> sp_sjobcontrol @name=NULL, @option="start_js"
2> go
```

5. Restart Adaptive Server if you restarted Job Scheduler manually.
6. (Optional) Add more log space. Some 64-bit platforms require additional space for the `sybmgmtdb` log:

```
1> use master
2> go
1> alter database sybmgmtdb LOG on sybmgmtdev=20
2> go
```

7. To upgrade `sybmgmtdb`, run the `installjsdb` script that is included, saving the output to a file:

   For UNIX:

```
isql -Usa -Psa_password -Sservername -n -i$SYBASE/$SYBASE_ASE/
scripts/installjsdb
-ooutput_file
```

   For Windows:

```
isql -Usa -Psa_password -Sservername -n -i%SYBASE%\%SYBASE_ASE%
\scripts\installjsdb -ooutput_file
```

Note: When upgrading from Adaptive Server version 12.5.x to 15.5 and later, increase the size of `sybmgmtdb` from 50MB to at least 160MB. `sybmgmtdb` must have free log space of at least 60MB.