PUBLIC

2018-01-26

SAP Applications on SAP Adaptive Server Enterprise - Best Practices for Migration and Runtime

Content

1 Document History

The following table provides an overview on the most important document changes:

Version:	Date:	Description:
1.0	2012-12-20	Initial Version
1.1	2013-03-01	Formal Update
1.2	2014-04-30	Updated for SAP Sybase ASE 15.7 SP110
1.3	2015-04-30	Updated for SAP Sybase ASE 15.7 SP132 and SAP ASE 16.0 SP01 Optimized document structure by separating migration- and runtime-related topics.
1.4	2015-09-10	Update for SAP Business Warehouse
1.5	2016-09-29	Updated version
1.6	2018-01-15	Correction of links and updates

Note

Most recent version of this best practice guide is always attached to SAP Note 1680803.

2 Introduction

This document describes best practices that are specific to the migration and operation of SAP NetWeaver-based products on SAP Adaptive Server Enterprise (SAP ASE).

In addition, this guide provides best practices for SAP Business Warehouse (SAP BW) and other SAP applications based on SAP BW, like SCM/APO, SEM, and the SAP Solution Manager.

Besides the SAP ASE-specific migration optimizations described in this document, there are several database-independent tuning measures that should be considered:

- Sorted and Unsorted Unload
- Package Splitting
- Table Splitting
- R3load Options
- Migration and Distribution Monitor
- Defining Unload and Load Order

2.1 Prerequisites

You should be familiar with performing basic OS administration tasks, understand the basic architecture of SAP ASE and know how to use the ASE SQL command line interface (isql).

For the migration-related chapters, you should also be familiar with the database independent migration process as well as the SAP tools involved (R3load, Migration Monitor, Distribution Monitor and so on). The methods described can be used to optimize the standard system copy procedure.

The methods and procedures described in this document are dependent on specific enhancements that were provided through various updates of both SAP and ASE binaries. While there are mandatory minimums, because product defects have been fixed in later releases as well as new functionality introduced, it is strongly recommended that you begin with a current supported version of the products and utilities.

It is not recommended, that you apply described methods to versions earlier than:

SAP ASE	SAP ASE 15.7 FOR BUS.SUITE SP137 SAP ASE 16.0 FOR BUS.SUITE SP02 PL04 HF1
ODBC	SAP ASE 15.7 FOR BUS. SUITE DBCLIENT.SAR SP21 PL0 SAP ASE 16.0 FOR BUS. SUITE DBCLIENT.SAR SP02 PL04 HF1/EBF 24040

See SAP Note 1554717 for up-to-date information on mandatory minimum versions. We recommend to stay on a recent patch level of SAP ASE, database client software, and SAP kernel to avoid problems due to software defects, which have been fixed in a later version.

Earlier installations to previous releases should consider upgrading when possible in order to leverage the new functionality and improvements.

SAP NetWeaver and SAP Business Suite customers are only allowed to download and use ASE binaries available in the SAP Software Center that are tagged *FOR BUS. SUITE*.

https://launchpad.support.sap.com/#/softwarecenter *Databases* > *Database and Database Patches* > *SAP Adaptive Server Enterprise* > *Database Patches* > *SAP ASE for Business Suite*

The Software Provisioning Manager is available on http://support.sap.com/sltoolset *System Provisioning* > *Installation Option of Software Provisioning Manager* > *Installation Guides - Application Server Systems* > *SAP Application Server Systems Based on SAP NetWeaver*

Select the database and product release via the filter function.

Refer also to the following SAP Notes:

1590719: *SYB: Updates for SAP Adaptive Server Enterprise (SAP ASE)*

1982469: *SYB: Updating SAP ASE with* `saphostctrl / sapdbctrl`

1607816: *SYB: Installing Service Packs for SAP ASE 15.7 (Windows)*

1599814: *SYB: Installing Service Packs for SAP ASE 15.7 (UNIX + Linux)*

2.2 Prerequisites for SAP BW

For SAP BW, SAP strongly recommends the use of a recent patch level of SAP ASE 16.0. This SAP ASE 16.0 provides several optimizations and features, especially for SAP BW:

- Intra-query parallelism on partitioned SAP BW tables
- Enhanced fact table hint support
- SAP BW-specific optimization with Online Analytical Processing (OLAP)
- Support for hyper-partitioned tables

3 Operating System Configuration

This section describes the configuration of the UNIX operating systems:

- Configuration for Linux [page 8]
- Configuration for HP-UX [page 11]
- Configuration for AIX [page 11]

3.1 Prerequisites

The following requirements apply to all UNIX operating systems:

- Set the *keepalive* interval. For more information, see SAP Note 1410736: *TCP/IP: Setting the keepalive interval*
- SAP ASE data and log devices on NFS mounted file systems should only be used with a network appliance filer. SAP ASE supports the database devices on NFS- and CIFS-mounted devices with network appliance filers for storing data.
  For more information, see the SAP ASE Configuration Guide https://help.sap.com/viewer/p/SAP_ASE *Configuration* › *Configuration Guide for UNIX* › *SAP ASE Devices and System Databases* › *Determine the Size of a Database Device* › *Support for Raw Partitions or Files*

3.2 Configuration for Linux

The following operating system tuning suggestions are recommended for Linux hosted systems:

- IO Subsystem Tuning [page 8]
- Memory Configuration and Tuning [page 9]
- Network Configuration/Tuning [page 11]

3.2.1 IO Subsystem Tuning

The IO subsystem as provided by Linux is by default tuned for desktop systems using non-enterprise storage, specifically low-end local disks and cached file system buffered reads and writes. This is completely the opposite of a typical SAP system which will use enterprise-class storage and extensive unbuffered access to storage devices. As a result, the following recommendations are strongly suggested for performance and stability:

- Use the `noop` IO scheduler (strongly recommended). If you do not change this for the entire system using the boot loader (for example, GRUB), you will need to add the change to the `rc.local` or `/etc/init.d/boot.local` to ensure that device settings are re-established on reboot.
- Increase the request queue for the storage block devices to *1024* by altering `/sys/block/<device>/queue/nr_requests`. As this change is not persistent, you will need to also add it to the `rc.local` or `/etc/init.d/boot.local` boot scripts.
- Make sure the kernel configuration `fs.aio-max-nr = 1048576` has been set. In addition, make sure that `fs.file-max` has been set in accordance with the aggregated requirements for all processing (recommendation is *6291456* as a minimum).
- For DBMS data storage, the `ext4` or `xfs` file systems are preferred over `ext3` as either can handle large files better than `ext3` (`ext4` & `xfs` use extents vs. `ext3`'s indirect blocks for allocations in large files among other benefits). For best performance, disable the journal via `tune2fs ^has_journal` and use the `noatime`, `nodiratime` mount options.
- If journaling is not disabled, for `ext4` use `cache=writeback,barrier=0` mount options in addition to `noatime`, `nodiratime`. These options ensure that only meta-data changes to the file system are journaled instead of both meta-data and data changes.
- For `xfs` the recommended mount options are `noatime,nodiratime`, `nobarrier`, `logbufs=8`.

3.2.2 Memory Configuration and Tuning

SAP ASE needs shared memory in order to communicate between various processes in the DBMS environment and the SAP processes in the central instance may have as well significant memory requirements. Follow the steps outlined below:

- Check whether `kernel.shmall` and `kernel.shmmax` are sufficient for each instance of SAP ASE and other processes using shared memory.
- Apply Linux paging improvements (see SAP Note 1557506).
- Set `vm.swappiness` to a reduced value such as *20* or *10* (vs. the default of *60*).
- Configure just the appropriate amount of huge pages necessary to support database memory requirements. Do not under or over-configure huge pages. In order for this change to work, you will likely need to modify `/etc/security/limits.conf` and add the DBMS user with `memlock` permission.
- Disable transparent huge pages.

> **Note**
>
> If using Linux kernels prior to 2.6.18-238, do not use huge pages. Prior to that kernel release, there was a bug in RedHat Enterprise Linux (593131) that did not release huge pages until a box reboot.

Huge pages may help to decrease OS overhead by decreasing the page table size. However, changing the number and size of huge pages may require a reboot of the operating system, and thus could be unsuitable if SAP ASE needs to be reconfigured frequently. Utilizing huge pages does provide measurable performance gains and is required if the DBMS memory will exceed 256 GB. Remember, in many cases, the SAP central instance is likely running on the same host as the DBMS which could cause memory contention if huge pages are over-configured.

The recommendation is to configure huge pages in even multiples of 256MB and only configure as many huge pages as necessary to support SAP ASE and any minor configuration changes. For example, if configuring SAP

ASE for 350GB of memory, you may wish to allocate 355 or 360GB in huge pages to allow for dynamically increasing SAP ASE memory in case configuration changes need to be made. Be extremely careful as over-configuring or under-configuring huge pages may degrade performance of SAP central instance components running on the same host. When it boots, SAP ASE attempts to grab all of its configured shared memory in a single memory allocation. If huge pages are under-configured, it will be forced to use normal-sized memory pages instead of huge pages. Since huge pages are locked in memory and can only be used by processes that request them, the amount of memory in huge pages that SAP ASE would have used is now unavailable to most applications. In addition, SAP ASE is using memory those applications would have used – a double memory hit. Similarly, if huge pages are over-configured, the amount of memory over-configured in huge pages will also be unavailable for most applications, which also could lead to excessive swapping.

One way to help prevent SAP ASE from contending for memory with SAP applications is to ensure that it only uses the huge pages defined (vs. normal memory pages if huge pages are not available). This can be configured in SAP ASE 15.7 SP 110 or higher by using the configuration parameter below:

```
--Ensure that we will only use huge pages on boot
exec sp_configure 'enable HugePages',2
go
```

Once configured, you will need to restart SAP ASE. If huge pages are not available, SAP ASE will not start. To ensure you have enough huge pages prior to restarting SAP ASE, issue the following command:

```
$ cat /proc/meminfo
MemTotal:       529182764 kB
MemFree:           418712 kB
Buffers:            30528 kB
Cached:          74129068 kB
SwapCached:       2666984 kB
...
HugePages Total:    38771
HugePages Free:     38771
HugePages_Rsvd:         0
HugePages_Surp:         0
Hugepagesize:        2048 kB
...
```

For example, the above only shows 38771 2MB huge pages available for a total of 77542MB (or ~75GB). This would not be sufficient to restart SAP ASE, if it was configured for 350GB. If nothing else is on the box and SAP ASE was running previously, you may need to check with `ipcs` if the shared memory segments are still in the operating system and if so, remove them with `ipcrm`.

When huge pages are in use, the total number needs to be obtained by subtracting both *free* and *reserved* from the *total*. The total huge pages in use would be 131072 (HugePages_Total) – 98409 (HugePages_Free) + 38874 (HugePages_Rsvd) = 131072-98409+38874=71537.

This equates to 143,074MB using 2MB huge pages (processor dependent sizing):

```
$ cat /proc/meminfo
...
HugePages_Total: 131072
HugePages_Free:   98409
HugePages_Rsvd:   34874
HugePages_Surp:       0
```

3.2.3 Network Configuration/Tuning

Due to unknown hardware at installation time, the core network configurations and tuning for Linux is minimal due to the amount of memory necessary which may not be available in small footprint environments. Because of the high amount of network traffic between SAP applications and the DBMS or between the SAP clients and the central instance server, tuning the network is strongly recommended. Like IO tuning, some of the configurations are changed in the OS kernel and others require changing the network device.

- Check whether TCP send/receive buffers (`net.core.rmem_max`, `net.core.wmem_max`, `net.ipv4.tcp_wmem`, `net.ipv4.tcp_rmem`, etc.) are large enough to support SAP ASE requirements. The defaults of 1MB are likely too small and a better starting position might be 16MB or higher.
- Consider increasing the length of the processor receive queue (`net.core.netdev_max_backlog` = 30000), as well as the transmit queue (`ifconfig <eth#> txqueuelen <value>`). For the latter, a value of 10000 is suggested as a starting point for 10GbE.

3.3 Configuration for HP-UX

For more information, see the following SAP Notes:

- SAP Note 837670: *Minimum OS patch recommendations for HP-UX*
- SAP Note 172747: *SAP on HP-UX: OS kernel parameter recommendations* and supplement with the file system cache tuning referenced in SAP Note 1077887.

3.4 Configuration for AIX

For SAP ASE specific tuning on IBM AIX, see the white paper Optimizing SAP Sybase Adaptive Server Enterprise (ASE) for IBM AIX which is available on the IBM web site. This paper discusses a number of topics that are common to IBM Power series server configurations and OS considerations including LPARs, Virtualized IO Servers (vios) and other aspects that may differ from site to site.

In addition to that paper, if the SAP installation includes the HADR solution with RMA and SAP Replication Server (SRS), on the hosts where SRS is running, make sure that the following environment variables are set by adding them to the appropriate `SRS RUNSERVER` file:

```
export AIXTHREAD_MUTEX_FAST="ON"
export YIEDLOOPTIME=0
export SPINLOOPTIME=500
export MALLOCOPTIONS="threadcache, multiheap:8, considersize"
```

Note that the `multiheap:8` is an example of creating 8 buckets. The actual number of buckets you should use depends on the number of threads SRS is using, which can be closely approximated by the number of inbound and outbound connections. For most SAP installations, there will only be connections for the master database and SID database as well as the RSSD and route queue for local topology - for a total of *4*. For the remote topology, the SRS will have both inbound and outbound connections for each database plus the RSSD for *5* connections. Consequently, the minimum value that should be used is *4*, but *8* might be a better starting point. The maximum value is *32*. Note that increasing the number of heap memory buckets may increase the process size in the operating systems as each heap memory allocation pool has a minimum size.

Consequently, increasing beyond what is needed may cause the SRS process size to be much larger than the SRS memory configuration values might suggest. Contact SAP support for further tuning and monitoring information.

4 Database Migration

4.1 General Considerations for Export and Import

- The top 10-25 tables (by size) should be split into 10-50 packages during export. You may extend this to even more tables.
- The Unicode conversion process requires that certain database tables be exported in a sorted manner. Tables such as `RFBLG` and other cluster tables are examples of such tables. `R3load` will automatically export tables that require sorted export in the correct mode, even if the default export setting has been set to *unsorted*.
- SAP ASE can make use of sorted data when creating the primary index. This, however, only applies to tables that have been exported sorted and have not used table splitting. Recommendation is to export all packages unsorted, unless being forced to sorted export due to reasons documented in SAP Note 954268. See also chapter DDL Template File Mapping [page 27] for more details.
- Generally, the export of packages is in alphabetical order. You may wish to customize this to start several of the exports on the larger tables first while concurrently exporting smaller tables.
- Overall import time is dominated by index creation. As a result, for larger installations, the index creation step should be sequenced separately by following the steps described in this document.
- The use of Solid State Disks (SSDs) can greatly reduce the export/import times, as well as improve the performance of the system after migration. SSDs should be considered for:
    - the DBMS devices (specifically the devices used for the log segment)
    - the export/import file system space
    - a snap copy of the system to be migrated

4.2 Distribution Monitor versus Migration Monitor

There are two different approaches to the migration. The first is a centralized approach in which the import/export will occur on the same host as the ASE instance. This centralized approach will use the `migmon` utility for the migration and generally should use a staged approach for the migration.

- Phase 1: Create table and load data using a high number of parallel import jobs, but reduced SAP ASE resources.
- Phase 2: Index and view creation, using a reduced number of parallel import jobs, but increased SAP ASE resources.

In a distributed migration, other hosts such as those used for SAP application servers, are used to split the import/export across multiple nodes and using the `distmon` SAP migration utility. In this case, the SAP ASE configuration can be set for the index creation levels with increased CPU and memory to support greater parallelism than the centralized approach would allow. This approach can be simpler as it allows the normal table import to also control the index creation as part of the load step vs. restarting the installation after reconfiguring the DBMS.

While some may consider having the import/exports local to the SAP ASE and using `migmon` as an advantage due to eliminating any network bottlenecks during the data import, generally speaking, the import process is extremely fast and the bulk of the time for the migration will be spent on creating the indexes. As a consequence, the time consumption for transferring imports across the network is more than compensated by using `distmon` and increasing the parallel activities.

Most of the advice in this document is the same no matter which approach is used. Where there are differences, it will be noted.

4.3 Preparations for SAP BW

Make sure, that you install a recent patch level of SAP ASE 16.0, when setting up a SAP BW system on SAP ASE. Do not use SAP ASE 15.7 for SAP BW systems.

In order to achieve optimum performance on different database platforms, SAP BW uses special database-specific features that cannot easily be mapped during migration and are not explicitly represented in the ABAP Dictionary. Therefore, it is necessary to perform additional steps during the migration of an SAP BW system.

> i Note
>
> The following recommendations also apply to SAP applications based on SAP BW, such as the SAP Solution Manager, SCM/APO, and SEM.

Carefully follow the pre-migration instructions in the following SAP Notes:

SAP Note 888210: *NW 7.**: System copy (supplementary note)*

SAP Note 2103871: *Migrating SAP BW to SAP Adaptive Server Enterprise*

To ensure the proper generation of DDL files during the export of the source system, apply the patch collection notes relevant for your SAP NetWeaver release **before** starting the export:

SAP Note 1605169: *SAP BW 7.02 Correction Collection*

SAP Note 1608417: *SAP BW 7.3 Correction Collection*

SAP Note 1616726: *SYB: SAP BW 7.31 Correction Collection*

SAP Note 1821924: *SYB: SAP BW 7.4 Correction Collection*

SAP Note 2193724: *SYB: SAP BW 7.5 Correction Collection*

SAP Note 1946164: *DDIC patch collection for release 7.02*

SAP Note 1965664: *DDIC patch collection for release 7.3*

SAP Note 1965754: *DDIC patch collection for release 7.31*

SAP Note 1965755: *DDIC patch collection for release 7.4*

As the DDL for tables and indexes specific to SAP BW will be pre-generated during the export, it is important to ensure the use of consumers in these DDL files. For more information, see SAP Note 1952189: *Parallel Index Creation with SMIGR_CREATE_DDL*.

The default value for the PLL degree is three (3). This default can be changed using the RSADMIN parameter SYBASE_CRE_INDEX_CONSUMERS, provided that enough database resources are available. RSADMIN parameters can be set with the help of report SAP_RSADMIN_MAINTAIN before the execution of SMIGR_CREATE_DDL.

Changes to the template file DDLSYB_LRG.TPL will not affect the migration of special SAP BW tables; see SAP Note 1680803. The generation of these tables and their indexes will only be controlled by the generated SQL files.

Use report RS_BW_PRE_MIGRATION (SAP Note 1690674) to identify potential issues caused either by a very high number of partitions on a table in the source DBMS or by inconsistent metadata.

Also check for orphaned or empty partitions in F fact tables. These can also be identified with the help of report RS_BW_PRE_MIGRATION in the source system. Use report SAP_DROP_EMPTY_FPARTITIONS to drop such (empty) partitions.

Database migrations with SAP ASE as the target database require special attention if F fact tables are to be created without partitions in ASE. Perform the additional configuration steps in the source system as described in SAP Note 1691300: *Unpartitioned F fact tables for InfoCubes*

ℹ Note

SMIGR_CREATE_DDL will generate warnings if the number of data or index partitions violates a certain threshold. The thresholds can be changed with RSADMIN parameters. Default numbers are:

SYBASE_THRESHOLD_PTNS = 1000

SYBASE_THRESHOLD_INDXPTNS = 15000

The CREATE TABLE and CREATE INDEX statements will be generated, but the warnings should not be ignored.

The thresholds define a comfort zone. SAP ASE 16.0 can manage up to a few thousand partitions with minor performance degradations. A database migration is possible even with a few thousand partitions. Approximately 4000 partitions should be considered as a maximum.

4.4 Import Preparation

4.4.1 DBMS File Systems

All data and log devices of SAP ASE must be located under /sybase/<SAPSID>/ (UNIX and Linux) or <Drive>:\sybase\<SAPSID> (Windows). If the installer does not find existing sapdata_* mount points, it will create a single `sapdata_1` subdirectory for all data devices of the SAP database, as well as `saplog_1` for the log device of the SAP database. If you want to distribute the log and data devices of the SAP database onto different file systems, you need to create the respective mount points in advance.

Example:

`/sybase/<SAPSID>/sapdata_1` (250 GB)

`/sybase/<SAPSID>/sapdata_2` (250 GB)

`/sybase/<SAPSID>/sapdata_3` (250 GB)

`/sybase/<SAPSID>/sapdata_4` (250 GB)

`/sybase/<SAPSID>/saplog_1` (50 GB)

In addition to the `sapdata_1-N` mount points, the following subdirectories will contain data or log devices of other databases created during installation:

`/sybase/<SAPSID>/sybsystem` (used for all system databases)

`/sybase/<SAPSID>/sybtemp` (used for the system temporary database)

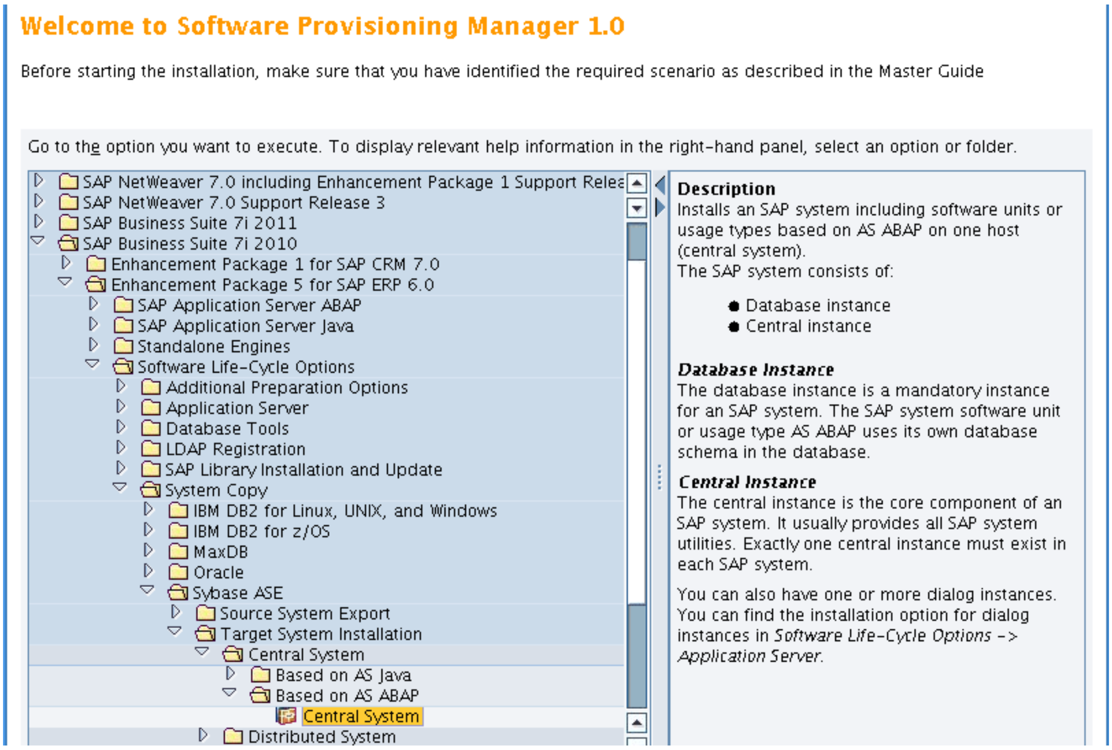
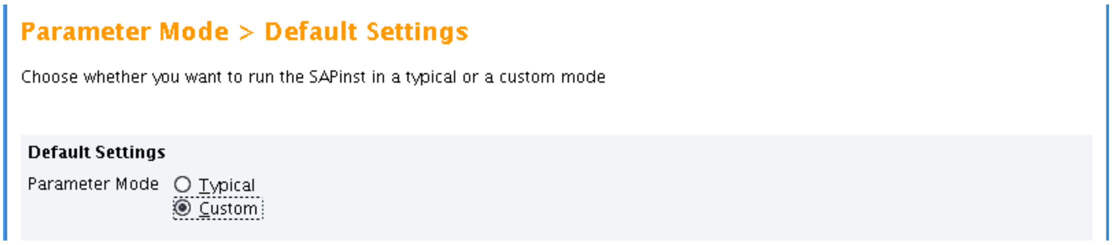
`/sybase/<SAPSID>/saptemp` (used for the SAP temporary database)

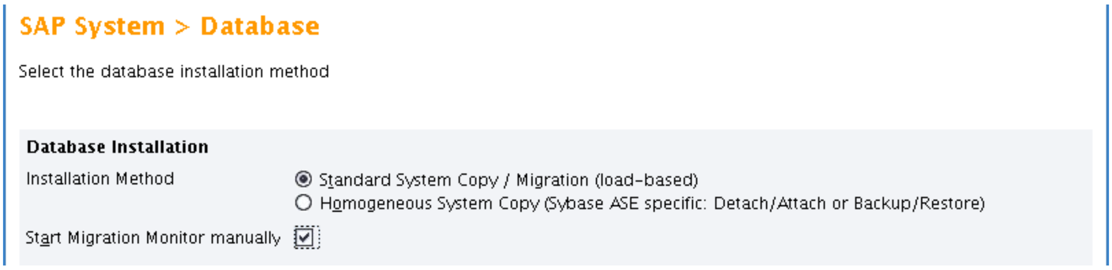
`/sybase/<SAPSID>/sapdiag` (used for the SAP monitoring repository / saptools database)

> i Note
>
> If SSD disks are available, the recommendation is to use them primarily for `saplog_1-N` as well as `saptemp`.

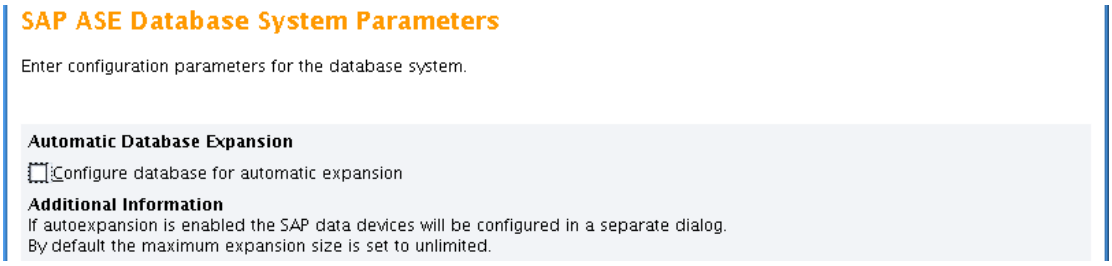
4.4.2 SAP and DBMS Software Installation

Perform the following steps:

Step	Description
1.	Start the Software Provisioning Manager and choose the appropriate installation option.  **Welcome to Software Provisioning Manager 1.0** Before starting the installation, make sure that you have identified the required scenario as described in the Master Guide Go to the option you want to execute. To display relevant help information in the right-hand panel, select an option or folder. SAP NetWeaver 7.0 including Enhancement Package 1 Support Relea SAP NetWeaver 7.0 Support Release 3 SAP Business Suite 7i 2011 SAP Business Suite 7i 2010 Enhancement Package 1 for SAP CRM 7.0 Enhancement Package 5 for SAP ERP 6.0 SAP Application Server ABAP SAP Application Server Java Standalone Engines Software Life-Cycle Options Additional Preparation Options Application Server Database Tools LDAP Registration SAP Library Installation and Update System Copy IBM DB2 for Linux, UNIX, and Windows IBM DB2 for z/OS MaxDB Oracle Sybase ASE Source System Export Target System Installation Central System Based on AS Java Based on AS ABAP Central System Distributed System **Description** Installs an SAP system including software units or usage types based on AS ABAP on one host (central system). The SAP system consists of: ● Database instance ● Central instance ***Database Instance*** The database instance is a mandatory instance for an SAP system. The SAP system software unit or usage type AS ABAP uses its own database schema in the database. ***Central Instance*** The central instance is the core component of an SAP system. It usually provides all SAP system utilities. Exactly one central instance must exist in each SAP system. You can also have one or more dialog instances. You can find the installation option for dialog instances in *Software Life-Cycle Options -> Application Server*.
2.	Choose *Custom* installation mode.  **Parameter Mode > Default Settings** Choose whether you want to run the SAPinst in a typical or a custom mode **Default Settings** Parameter Mode ○ Typical ● Custom
3.	Specify the `SAP system ID, FQDN`, and user passwords as usual. ℹ Note As DBA Cockpit for SAP ASE is completely based on SAP Web Dynpro, it is strongly recommended that you set up the `FQDN` during installation of the SAP system.

Step	Description
4.	Choose *Standard System Copy / Migration (load-based)* and *Start Migration Monitor Manually*.  SAP System > Database Select the database installation method **Database Installation** Installation Method ● Standard System Copy / Migration (load-based) ○ Homogeneous System Copy (Sybase ASE specific: Detach/Attach or Backup/Restore) Start Migration Monitor manually ☑
5.	Specify the user passwords and the media and export locations as usual.

Step	Description
6.	Deselect the use of *Automatic Database Expansion*:



SAP ASE Database System Parameters

Enter configuration parameters for the database system.

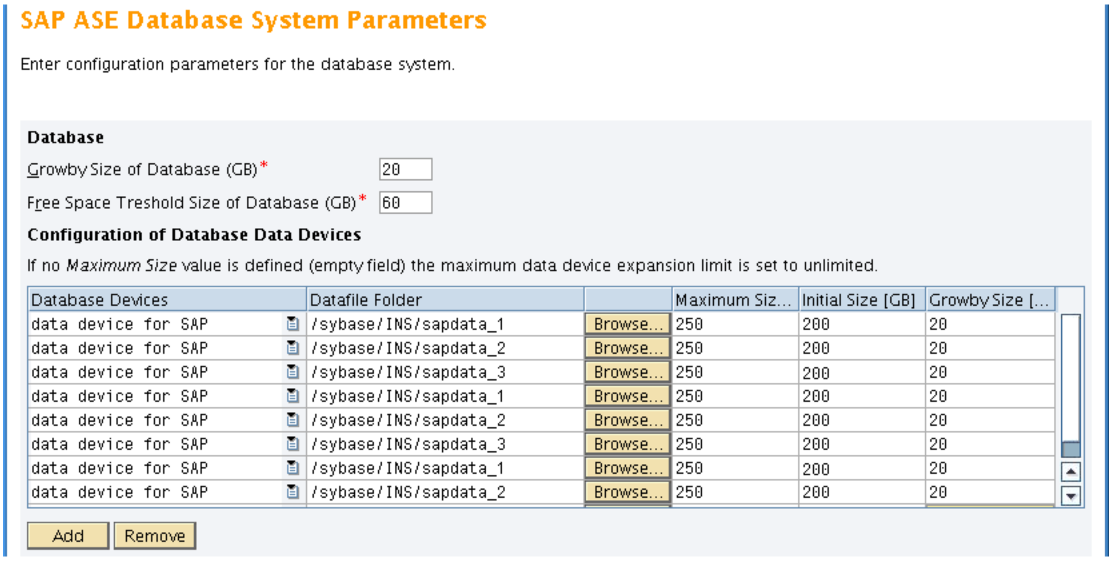
Automatic Database Expansion

□ Configure database for automatic expansion

Additional Information

If autoexpansion is enabled the SAP data devices will be configured in a separate dialog.
By default the maximum expansion size is set to unlimited.

> **Note**
>
> Automatic database expansion is a great feature to limit DBA interaction on continuously growing databases. However, during initial load of an SAP database, the continuous expansion of devices is usually not quick enough to keep up with the speed of data being imported using the *fast* mode of `R3load`.
>
> One option to still make use automatic database expansion during migration is to use reasonable initial sizes for all the ASE data devices. If for example you estimate a target database size of 2TB, you may configure nine ASE devices with an initial size of 200GB each (= 1.8TB in total), and let automatic database expansion resize the database to a maximum of 9x 250GB (= 2.25TB in total) if needed. Important hereby is to set *Free Space Threshold Size* to at least three times the *Growby Size*:



SAP ASE Database System Parameters

Enter configuration parameters for the database system.

Database

Growby Size of Database (GB)* 20

Free Space Treshold Size of Database (GB)* 60

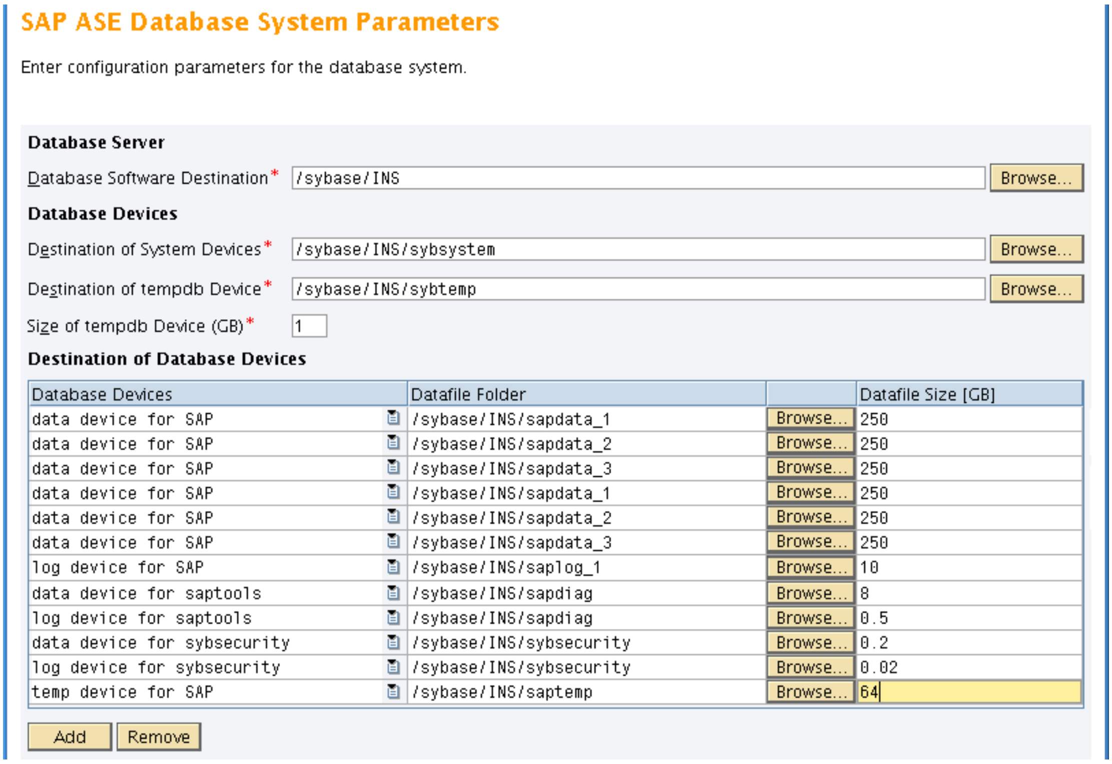
Configuration of Database Data Devices

If no *Maximum Size* value is defined (empty field) the maximum data device expansion limit is set to unlimited.

Database Devices	Datafile Folder		Maximum Siz...	Initial Size [GB]	Growby Size [...
data device for SAP	/sybase/INS/sapdata_1	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_2	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_3	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_1	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_2	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_3	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_1	Browse...	250	200	20
data device for SAP	/sybase/INS/sapdata_2	Browse...	250	200	20

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>7.</td><td>
<p>Data Devices for SAP Database:</p>
<ul>
<li>Use at least 4 devices. Do not use files larger than 250GB.</li>
<li>Estimation of target database size:</li>
</ul>
<table>
<tr><th></th><th>Source Uncompressed</th><th>Source Data Compressed</th><th>Source Data + Index Compressed</th></tr>
<tr><td><b>ASE 15.7 / ASE 16.0 without Index Compression</b></td><td>70% of source DB</td><td>100% of source DB</td><td>120% of source DB</td></tr>
<tr><td><b>ASE 16.0 with Index Compression</b></td><td>60% of source DB</td><td>80-90% of source DB</td><td>100% of source DB</td></tr>
</table>
<p>See also Extending the Database Size [page 25].</p>
<p>Log Device for SAP Database:</p>
<ul>
<li>Use log size of source DBMS.</li>
</ul>
<p>Monitoring Repository <code>saptools</code>:</p>
<ul>
<li>Depending on system size and intended retention period for monitoring data, allocate 5 - 20GB for data, and 0.5 - 1GB for log.</li>
</ul>
<p>Temporary Databases:</p>
<ul>
<li>As all logins, with the exception of <code>sapsso</code>, are bound to <code>saptempdb</code>, a default <code>tempdb</code> size of 1 GB is sufficient.</li>
<li>Size of <code>saptempdb</code> is similar to the space you reserved for temporary data in your source database. As an alternative, the following formula provides a good starting point: <i>square root (total DB size in GB) * 2</i></li>
</ul>
</td></tr>
</table>

Step	Description



SAP ASE Database System Parameters

Enter configuration parameters for the database system.

Database Server

Database Software Destination*	/sybase/INS	Browse...

Database Devices

Destination of System Devices*	/sybase/INS/sybsystem	Browse...
Destination of tempdb Device*	/sybase/INS/sybtemp	Browse...
Size of tempdb Device (GB)*	1	

Destination of Database Devices

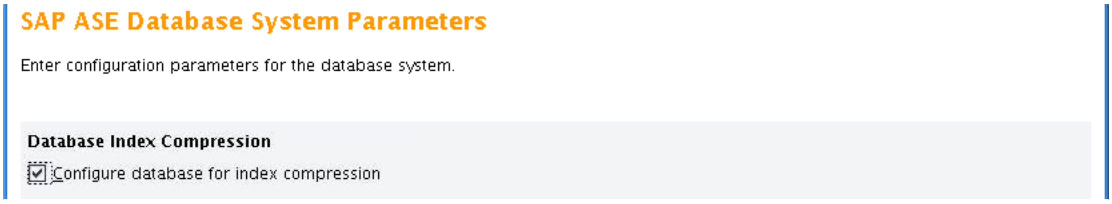
Database Devices	Datafile Folder		Datafile Size [GB]
data device for SAP	/sybase/INS/sapdata_1	Browse...	250
data device for SAP	/sybase/INS/sapdata_2	Browse...	250
data device for SAP	/sybase/INS/sapdata_3	Browse...	250
data device for SAP	/sybase/INS/sapdata_1	Browse...	250
data device for SAP	/sybase/INS/sapdata_2	Browse...	250
data device for SAP	/sybase/INS/sapdata_3	Browse...	250
log device for SAP	/sybase/INS/saplog_1	Browse...	10
data device for saptools	/sybase/INS/sapdiag	Browse...	8
log device for saptools	/sybase/INS/sapdiag	Browse...	0.5
data device for sybsecurity	/sybase/INS/sybsecurity	Browse...	0.2
log device for sybsecurity	/sybase/INS/sybsecurity	Browse...	0.02
temp device for SAP	/sybase/INS/saptemp	Browse...	64

Note

- SAP ASE devices will be created and allocated to the SAP database in the order listed. To spread I/O around a bit, it is recommended that you follow a “round-robin” approach instead of allocating devices on the same OS file system in sequence.
- The `sapdata_[1-N]` file systems should have sufficient space so that all devices fit within 80-85% of the available file system space to avoid free space search overhead – or the file system may need tuning to reduce the free space overhead.
- The installer will block you from continuing if the calculated minimal database size required is larger than the sum of all devices specified. To manually adjust the minimal database size, modify the `DBSIZE.XML` file, which can be found in `<export directory>/DB/SYB`.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>8.</td><td>

As of Software Provisioning Manager 1.0 SP8 and ASE 16.0, you can choose to make use of index compression:



SAP ASE Database System Parameters

Enter configuration parameters for the database system.

Database Index Compression

☑ Configure database for index compression

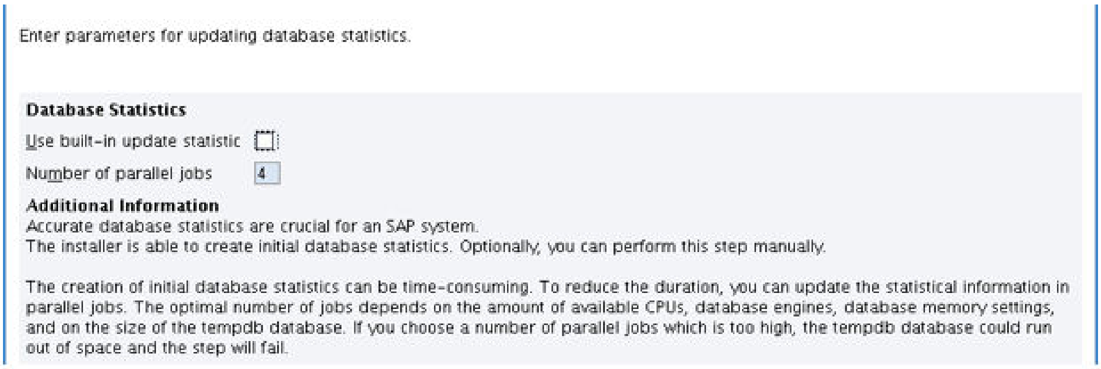
Note

If this option is not offered by the installer being used, an alternative is to enable index compression manually before starting the import:

```
use master
go
alter database <SID> set index_compression = page
go
```

</td></tr>
<tr><td>9.</td><td>Confirm ASE ports, user passwords, and ASE resource allocation. Resource allocation will be adjusted in the following steps, so it is not relevant at this stage.</td></tr>
<tr><td>10.</td><td>

Uncheck the *Use built-in update statistics* option to let the installer know that the user wants to create table statistics manually.



Enter parameters for updating database statistics.

Database Statistics

Use built-in update statistic □

Number of parallel jobs 4

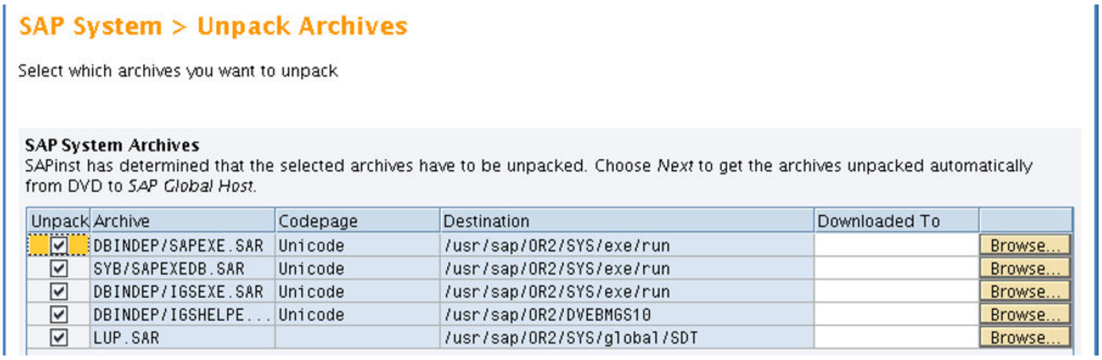
Additional Information

Accurate database statistics are crucial for an SAP system.
The installer is able to create initial database statistics. Optionally, you can perform this step manually.

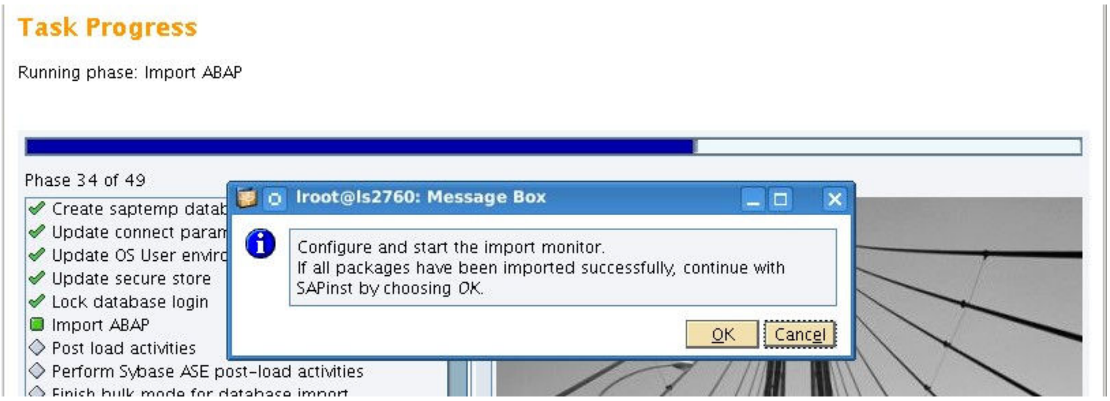
The creation of initial database statistics can be time-consuming. To reduce the duration, you can update the statistical information in parallel jobs. The optimal number of jobs depends on the amount of available CPUs, database engines, database memory settings, and on the size of the tempdb database. If you choose a number of parallel jobs which is too high, the tempdb database could run out of space and the step will fail.

</td></tr>
<tr><td>11.</td><td>Specify the SAP system parameters and user passwords as usual. It is not necessary to interrupt the installation before starting the system specific to the migration process described in this document.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>12.</td><td>You can ask the installer to apply the kernel updates during install of the SAP instance by specifying the download locations. Assuming use of recent SWPM version, this is usually not needed.



SAP System &gt; Unpack Archives
Select which archives you want to unpack
<b>SAP System Archives</b>
SAPinst has determined that the selected archives have to be unpacked. Choose <i>Next</i> to get the archives unpacked automatically from DVD to <i>SAP Global Host</i>.
<table>
<tr><th>Unpack</th><th>Archive</th><th>Codepage</th><th>Destination</th><th>Downloaded To</th><th></th></tr>
<tr><td>☑</td><td>DBINDEP/SAPEXE.SAR</td><td>Unicode</td><td>/usr/sap/OR2/SYS/exe/run</td><td></td><td>Browse...</td></tr>
<tr><td>☑</td><td>SYB/SAPEXEDB.SAR</td><td>Unicode</td><td>/usr/sap/OR2/SYS/exe/run</td><td></td><td>Browse...</td></tr>
<tr><td>☑</td><td>DBINDEP/IGSEXE.SAR</td><td>Unicode</td><td>/usr/sap/OR2/SYS/exe/run</td><td></td><td>Browse...</td></tr>
<tr><td>☑</td><td>DBINDEP/IGSHELPE...</td><td>Unicode</td><td>/usr/sap/OR2/DVEBMGS10</td><td></td><td>Browse...</td></tr>
<tr><td>☑</td><td>LUP.SAR</td><td></td><td>/usr/sap/OR2/SYS/global/SDT</td><td></td><td>Browse...</td></tr>
</table>
</td></tr>
<tr><td>13.</td><td>After rechecking all parameters, choose to start the installation process.
i Note
To speed up the step <i>Initialize database devices</i>, you can manually set the SAP ASE configuration parameter <i>number of large i/o buffers</i> to 64 after step <i>Verify database servers</i> has completed. This will speed up clearing the database pages during <code>CREATE/ALTER DATABASE</code>. Since login <code>sapsa</code> does not yet exist, you will need to log in as user <code>sa</code>.</td></tr>
<tr><td>14.</td><td>The installer will stop at step <i>ABAP Import</i> and wait until all packages have been imported manually.



Task Progress
Running phase: Import ABAP
Phase 34 of 49
✓ Create saptemp datab
✓ Update connect param
✓ Update OS User enviro
✓ Update secure store
✓ Lock database login
🟩 Import ABAP
◇ Post load activities
◇ Perform Sybase ASE post-load activities
◇ Finish bulk mode for database import
lroot@ls2760: Message Box
Configure and start the import monitor.
If all packages have been imported successfully, continue with SAPinst by choosing <i>OK</i>.

</td></tr>
</table>

4.4.3 SAP and DBMS Software Update

Before you start the import, it is important that you verify the SAP and DBMS binary versions and update the same if they do not match the minimum requirements described in section *Prerequisites* [page 6] .

Perform the following steps:

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1.</td><td>

Verify the version of the DB-independent part of the SAP kernel:

1. Execute `disp+work -V`
2. Ensure *kernel release* is 720 and *patch number* is at least 400.

```
--------------------
disp+work information
--------------------
kernel release     720
...
patch number       700
```

3. Update the DB-independent part of the SAP kernel, if necessary.

Note

At this stage of the installation, the SAP instance directory does not yet exist and thus `sapcpe` must not be executed after updating the SAP kernel in the `/sapmnt/<SAPSID>` directory.

</td></tr>
<tr><td>2.</td><td>

Verify the version of the DB-dependent part of the SAP kernel:

1. Execute `R3trans -d` and ignore return code *12*.
2. Open `trans.log` and search for the following line to ensure DBSL release is at least 720 patch level 610.

```
Version of '/usr/sap/OR2/SYS/exe/run/dbsybslib.so' is "720.00",
patchlevel (0.618)
```

3. Execute `R3load -V` and `R3ldctl -V`
4. Ensure both utilities have at least *kernel release* 720 and *source id* 0.610

```
--------------------
R3ldctl information
--------------------
kernel release    720
...
patch number      613
source id         0.613
```

```
--------------------
R3load information
--------------------
kernel release    720
...
patch number      716
source id         0.716
```

5. Update the DB-dependent parts of SAP kernel if necessary.

</td></tr>
<tr><td>3.</td><td>

Verify the version of SAP ASE:

1. Switch to user `syb<sapsid>`.
2. On Linux/UNIX, execute `dataserver --sbssav`
3. On Windows, execute `sqlsrvr.exe --sbssav`
4. Ensure that your SAP ASE version is a current version. See also SAP Note 1590719
5. If necessary, update SAP ASE by following the steps described in SAP Note 1982469.

</td></tr>
</table>

Step	Description
4.	Verify the version of the ASE ODBC driver: 1. Execute `R3trans -d` and ignore return code *12*. 2. Open `trans.log` and search for the following line to ensure the ASE ODBC driver release is at least 15.7.0.1260 or 16.0.01.00: `Driver: Adaptive Server Enterprise (ANSI) Driver release: **16.0.01.00**` 3. If necessary, update the ASE ODBC driver by following the steps described in SAP Note 1599814 (UNIX/Linux) or SAP Note 1607816 (Windows).

4.4.4 Extending the Database Size

In the case of very large (multi-TB) installations, due to the number of 250GB file devices necessary, you may find it tedious to create all the database space using the installation tool. You can create the initial database using a single data device and then extend the database to its full size using the `disk init` and `alter database` commands. To do this more efficiently, check and make sure that the *number of large I/O buffers* is higher than 32. If not, set it using the following command:

```
-- This will speed up create / alter database while clearing database pages
exec sp_configure 'number of large i/o buffers', 64
go
```

Note

Each concurrent database load, alter database or create database can only use up to 32 large IO buffers. The above configuration will support two concurrent operations if necessary, even though only one operation at a time will be likely. However, if you extend both `saptempdb` and the SAP database at the same time, the above configuration will help.

4.4.5 Optimizing On-Row LOB Length and Compression Settings

Some tables are excluded from compression by `R3load`. These are tables containing data that is already compressed by the SAP application (SAP pool and cluster tables), as well as very volatile tables like the VB- and RFC-queues.

For most of the tables containing LOB data, the SAP ABAP Dictionary defines the maximum LOB length. This information is automatically applied as the ASE on-row LOB length. For LOB columns that do not have a maximum length defined in the SAP ABAP Dictionary, or whose defined length is larger than 4000 bytes, an on-row LOB length of 4000 bytes is used.

In addition to the defaults used by R3load, there is an option to overrule compression settings as well as on-row LOB length for individual tables and table columns (see also SAP Note 1657918).

Examples:

- To disable compression for table SVERS, add the following line to DDLSYB.TPL / DDLSYB_LRG.TPL:

  ```
  negcpr: SVERS
  ```

- To modify the on-row LOB length for column DATA of table REPOSRC, add the following line to DDLSYB.TPL / DDLSYB_LRG.TPL:

  ```
  irll: REPOSRC DATA 16000
  ```

Download and unpack DDLSYB.SAR, which is attached to SAP Note 1680803. Replace DDLSYB.TPL and DDLSYB_LRG.TPL in <import_directory>/ABAP/DB/ with the files included in DDLSYB.SAR. This ensures that R3load uses the best known table storage settings.

To further optimize the on-row LOB length, specifically on large tables containing transactional data, you can execute report ZSAMPLELOBLEN on the source SAP system. Report ZSAMPLELOBLEN is attached to SAP Note 1680803 and samples LOB lengths for any table that has a potentially non-optimal default on-row LOB length. It is not necessary to increase the on-row LOB length to an extent that allows any LOB to fit on the data page, but you should try to get at least 75% of the LOBs on-row for the large transaction tables.

i Note

Large on-row LOB lengths will limit the number of rows that fit onto a single data page. This can have a negative impact on I/O in cases where the LOB portion of the row is rarely accessed. Although it might be tempting to move as much of the LOB data in-row as possible during migration, a sudden large increase in the row size might increase the number of pages that need to be accessed for non-LOB data. The best approach is to use the pre-migration or default settings as recommended by SAP and then monitor the system post-migration. The frequency of LOB access can be measured via monitoring master..monOpenObjectActivity for IndexID=255 (LOB) and then compared to the access for IndexID=0 (base table). If the frequency is high, consider increasing the LOB in-row size to reduce the off-row LOB fetching using the text chains. For example, consider the following result of a 10-hour observation:

ObjectName	IndexID	LogicalReads	PhysicalReads	APFReads	PagesRead	UsedCount
CIF_IMOD	0	128.322	1.140	9.653	8.245	14
tCIF_IMOD	255	316.437	13.030	0	104.177	0

Compared to the logical reads on the base table, the text data is accessed quite frequently. It could be because there are multiple text pages per entry and off-row storage is unavoidable. It also could be that each of the ~50,000 queries returned 6 rows on average with the text data. The only way to know for sure would be to use ZSAMPLELOBLEN and determine if the data can be moved in-row to speed up query access and possibly reduce the table's storage footprint.

4.5 Import

Whether using migmon or distmon, the optimal ratio of R3load processes and ASE threads is heavily dependent on the type of tasks that are primarily executed during certain phases of the import.

Background:

- Data, including LOB data, is loaded using the bulk copy feature of SAP ASE. With bulk copy, rows are completely formatted on the client side. This causes increased CPU and memory consumption on the client / R3load side during data load, while ASE is mostly a pass-through component doing I/O.
- Index creation, especially with the consumers option, consumes large amounts of CPU and memory on the ASE side to perform the necessary sort operations.
- To ensure that only minimal logging occurs, R3load creates indexes only after all data has been imported into a certain table.

Given this significant difference in resource requirements for data import and index creation, if using the centralized migmon or a composite migmon/distmon approach, the import phase can be optimized by staging the two tasks into two separate phases:

- Phase 1: Create table and load data using a high number of parallel import jobs, but reduced ASE resources.
- Phase 2: Index and view creation, using a reduced number of parallel import jobs, but increased ASE resources.

The following sections describe all the necessary steps for the import. While this is the general recommendation for the migration of multi-terabyte SAP systems, it should be considered optional for migrations of medium-sized systems.

4.5.1 DDL Template File Mapping

Two DDL template files are provided for DBMS type SAP ASE (SYB).

- For any table that has been exported unsorted, or that has been split into multiple packages during export, the template file DDLSYB_LRG.TPL must be used for import. DDLSYB_LRG.TPL will omit the sorted_data option during creation of the primary index, but make use of consumers to speed up the necessary sort.
- For any table that has been exported sorted, the template file DDLSYB.TPL should be used for import. The DDLSYB.TPL makes use of the sorted_data option during creation of the primary index, and thus avoids any unnecessary sort. However, depending on the source system, this requires a sorted export – which can significantly extend the export times depending on the source DBMS. If the export time increase is expected to be greater than the index sort times, it may be best to export unsorted and to use DDLSYB_LRG.TPL exclusively.
  See SAP Note 954268 regarding restrictions that apply for unsorted exports.
- Both DDL template files make use of 3 consumers for the creation of indexes.

Download and unpack DDLSYB.SAR, which is attached to SAP Note 1680803 . Replace DDLSYB.TPL and DDLSYB_LRG.TPL in <import_directory>/ABAP/DB/ with the files included in DDLSYB.SAR. This ensures that R3load uses the best known table storage settings. Additionally, these DDL template files are making use of the *with statistics* option during index creation, which will allow us to completely skip the *Update database statistics* phase of the SAP installer that is scheduled post import.

DDL template files can be mapped to `R3load` import packages using the parameter `ddlMap` of `import_monitor_cmd.properties`. See the Migration Monitor User's Guide for more information.

4.5.2 Optimizing the ASE Configuration for Data Import

Note that depending on the ASE specifications provided to the SAP installer, some of the configurations below may already be set. However, if you chose a default installation, you may need to do all the configurations listed in this section.

Memory Resources

Many of the internal resources used by different ASE configurations will need considerable amounts of memory. Consider the below table of configuration values, how to estimate memory use, the typical setting for a large SAP ERP system on ASE with 1000 concurrent users (2000 connections) and the amount of memory it likely would consume. Note that these values are close approximations only

Configuration Setting	Memory Estimation	Typical Value	Memory Consumption
number of locks	1.25GB per 5 million	10000000	2560MB
number of open objects	115MB per 50000	100000	230MB
number of open indexes	105MB per 50000	250000	525MB
number of open partitions	71MB per 50000	250000	355MB
compression info pool	32-64MB per 500 users	131072	256MB
large IO buffers	4MB each	64	256MB
number of user connections	512MB per 1000	2000	1024MB
heap memory per user	Global pool is based on this setting x number of user connections.	98304	256MB
kernel resource memory	SAP recommended setting	148279	300MB
			5762MB

As you can see, a large ERP installation may need ~6GB just for internal structures. Note that not all of these will be needed during data import – for example, the *number of open objects*, etc. are not as applicable during import as in runtime. In addition, you will need approximately 60-100MB of procedure cache per concurrent index creation – which likely is the number of concurrent `migmon` or `distmon` processes you will be using –

which is probably controlled by the number of ASE engines. The following command examples assume a total of 100GB physical memory available on the database host for the ASE instance a plan for running 30-40 concurrent import streams. You will need to make adjustments based on the amount of memory you have available.

```
use master
go
-- max memory sets the maximum physical memory that ASE can allocate from the
OS. It
-- should be set to the total OS physical memory, less 4-16GB for OS tasks and
-- management utilities, less X GB (where X corresponds to the number of parallel
-- R3load import jobs planned for the import phase).
-- Whatever the case, ensure ASE memory does not swap by checking other memory
-- consumers carefully.
exec sp_configure 'max memory', 0, '80G'
go
exec sp_configure 'allocate max shared memory', 1
exec sp_configure 'number of index trips', 0
exec sp_configure 'number of open databases', 20
go
-- This will speed up create / alter database while clearing database pages
exec sp_configure 'number of large i/o buffers', 64
go
-- Procedure cache is the main working memory of ASE. During import phase, it is
-- predominantly used for sorting (create index, update statistics).
-- Set it to 10-15% of max memory (with minimum of 512MB).    Each concurrent
create
-- index can use 60-100MB of proc cache, so we would need at least 4GB just for
-- sorting.    Best bet is to use a value 2.5x that or 10GB to allow room for
other
-- requirements.
exec sp_configure 'procedure cache size', 0, '10G'
go
-- Enable statistics hashing during both update statistics and create index.
Later
-- is important as it allows us to completely skip the 'Update database
statistics'
-- phase after the import.
exec sp_configure 'update statistics hashing', 0, 'on'
exec sp_configure 'utility statistics hashing', 0, 'on'
exec sp_configure 'number of histogram steps', 100
exec sp_configure 'number of sort buffers', 10000
go
-- Bulk load requires a lot of heap memory for each connection. However, heap
-- memory is a global pool from which any single user can allocate as much memory
-- as necessary for the current operation up to the pool size. The global pool
size
-- is defined by the 'heap memory per user' * 'number of user connections'
formula.
-- So as not run out of heap memory, especially if using more than 10 parallel
import
-- jobs, we need to increase the total available heap memory by increasing the
-- number of user connections.
exec sp_configure 'number of user connections', 500
go
-- configure sufficient compression memory for initial load
-- (256MB or 131072 2K pages)
exec sp_configure 'compression info pool size', 131072
go
```

CPU Resources

The following command examples assume a total of 32 physical cores available on the database host. During data import, ASE is essentially a pass-thru component. If using the staged approach with `migmon`, we will need to limit the number of ASE user threads to around one third of the number of physical cores to allow processing resources for R3load. If using a `distmon` and a fully distributed import, the amount of CPU resources can be set as it would be for index creation vs. limiting. Note that we are referring to physical cores and not logical processors (threads).

Max. Online Engines

The `max online engines` configuration parameter controls the maximum number of ASE query engines in the default and any user defined thread pools that can be brought online. It does not set the number of ASE engines that will be started when ASE boots.

We recommend to configure SAP ASE engines not higher than the number of physical cores available and dedicated for the SAP ASE Server. On a dedicated database host with 32 cores, the maximum number of engines you can configure is 32. A higher number of engines is not advisable.

```
use master
go
-- max online engines sets the maximum number of ASE engines that can be started
-- without restarting the ASE server. The actual number of cores/threads used
will
-- be controlled by the thread pools.
exec sp_configure 'max online engines', 32
go
```

Thread Pool Sizing

During the initial configuration, there should only be the three system supplied thread pools. The one primarily used for query processing will be the `syb_default_pool`. The number of engines required in this pool for data import may vary depending on the migration methodology. If using `migmon` and running `R3load` processes local to the database engine, you may want to restrict the `syb_default_pool` to 50% of the host capacity - for example 16 engines on a 32 core host. It might be best to start with 24 and dynamically increase to 32 by observing the CPU utilization of the `R3oad` processes and ASE engines. If using `distmon`, it may be possible to increase the number of engines to the full 48 on a 32 core/64 thread host.

```
use master
go
-- (Linux/UNIX only)
-- Normally, we would configure 1 network task per every 3-4 engines. However,
-- for data load, network responsiveness is not as critical as ASE will not be
-- sending much other than acknowledgements, so we can halve that.
if( @@version not like '%Windows%' )
begin
  exec sp_configure 'number of network tasks', 4
end
go
-- (Linux/UNIX only)
-- During data load, the IO subsystem will be very busy. Having 2 disk tasks will
-- ensure ASE can always keep up with a fast IO subsystem.
if( @@version not like '%Windows%' )
begin
  exec sp_configure 'number of disk tasks', 2
end
go
-- The actual number of OS cores utilized by ASE is determined by the number of
ASE
-- threads. By default, there are 3 groups of threads:
-- * User threads (syb_default_pool)
-- * I/O threads (syb_blocking_pool)
-- * System threads (syb_system_pool)
-- While syb_blocking_pool and syb_system_pool stay untouched, increase the
number
-- of threads in syb_default_pool to the number of CPU cores reserved for ASE
-- less 1 or 2 depending on the number of disk and network tasks. Make sure that
-- the default pool's idle timeout is 250.
-- In this case, we have 32 cores. However,
-- we will start with 24 an increase as needed if we decide to increase the
-- concurrency. We need to be careful, once the IO system is saturated,
increasing
-- ASE engines and import processes could lead to over-saturation and
degradation.
alter thread pool syb_default_pool with thread count = 24, idle timeout = 250
go
-- Configure 2.5x engines in worker processes and up to 25-50% of the engines as
-- the parallel degree for maximum utility parallelism – such as repartitioning,
-- etc. as well as parallel index creation although we will not be using that
-- many parallel threads per index.
exec sp_configure 'number of worker processes', 80
exec sp_configure 'max parallel degree', 12
exec sp_configure 'max utility parallel degree', 12
go
```

Database option for the SAP Database <DBSID>

During import phase the following database option must be set for the SAP database <DBSID>:

```
'select into/bulkcopy/pllsort' , 'true'
'trunc log on chkpt' , 'true'
'full logging for select into' ,  'false'
```

These options should have already been set by the SAP Installer.

SAP Temporary Database

The SAP installer automatically creates and configures a separate temporary database names `saptempdb`. Review its size and increase if necessary. If migrated from another database platform, recommendation is to size `saptempdb` similar to the source database. As an alternative, the following formula provides a good starting point:

*square root (total DB size in GB) * 2*

For example, a SAP database expected size of 5TB will need sqrt(5120)*2 = 71.55*2 = 143GB `saptempdb`.

In addition to the size, make sure that the following user bindings are established. This will prevent SAP processes (including R3load) from connecting to ASE prematurely during recovery should you need to reboot ASE for reconfiguration.

```
-- The following should already be set by SAPINST
-- Bind the SAP users to saptempdb
exec sp_tempdb 'bind', 'lg', 'SAPSR3', 'db', 'saptempdb', null, 'hard'
exec sp_tempdb 'bind', 'lg', 'SAPSR3DB', 'db', 'saptempdb', null, 'hard'
go
-- As sapsa is usually used to run update statistics on SAP objects, sapsa is
also
-- bound to the larger saptempdb. However, in case of sapsa, we default to a soft
-- binding so it can log in without saptempdb being available.
exec sp_tempdb 'bind', 'lg', 'sapsa', 'db', 'saptempdb'
go
```

Note that application users `SAPSR3` and `SAPSR3DB` have *hard* bindings which will ensure that they cannot login until `saptempdb` is online. However, for obvious reasons, the `sapsa` login uses the default *soft* binding. If `saptempdb` is not online, the `sapsa` login will connect using the default `tempdb`. If this happens (e.g. during recovery), you will need to be careful which commands you execute as `sapsa`. For example, commands that use a lot of temp space such as create index or update statistics may fail. Since a `tempdb` assignment is persistent for the duration of the session, in order to fix this if inadvertent, you will need to wait until `saptempdb` is recovered and then disconnect and reconnect `sapsa` logins.

If creating custom administrative logins in addition to `sapsa`, ensure those logins are also *soft* bound to `saptempdb`.

Locks

For most SAP systems, the default of one million locks configured by the installer will not be sufficient. If systems are migrated from database platforms that provide lock escalation (like DB2 or MaxDB), or from platforms that do not have a global lock list (like Oracle), it can become difficult to estimate the maximum number of locks required simultaneously. The recommendation is to start with a *number of locks* configured between 2 and 5 million with a rough formula of 1 million locks per terabyte of used data space.

An important aspect of data loading is to ensure that there are sufficient locks to support parallel imports into multiple tables. Keep in mind that the default schema for the SAP database is to use data rows locking – and that bulk loading is transactional in nature – even if the row modifications are not logged. As a result, you need to have sufficient locks for the number of rows loaded per table based on the commit frequency of the bulk load operation. The default installation also should set the row lock promotion to 2 billion rows. As a consequence, the loads will not escalate to a partition or table lock as normally would be expected. This is deliberate as concurrent loads into the same table (due to splits) could contend with one another if lock escalation occurred.

```
use master
go
-- Avoid running out of locks during bulk load.    Depending on the number of
parallel
-- imports planned, you will need 100000 locks for each parallel import plus
some for
-- miscellaneous activities.    Each 1 million locks will use about 235MB of
memory.
-- If we assume 30 parallel imports, we will need then a minimum of 3 million
locks
-- so, it is likely best to configure for 5000000 locks
exec sp_configure 'number of locks', 5000000
```

```
go
-- Generally, the lock hash table should be set to a maximum of a number of
locks/5
-- rounded down to the nearest power of 2. However, that could be rather high -
we
-- should configure this for the 90%-tile point of locks likely used when in
-- production as bulk loads will likely use all the available locks. Generally,
we
-- only see ~15% of configured locks in use at a time.
exec sp_configure 'lock hashtable size', 65536
go
-- The following should already be set by the SAP installer, but check
exec sp_configure 'row lock promotion LWM', 2147483647
exec sp_configure 'row lock promotion HWM', 2147483647
go
```

Default Data Cache

During data import, we will only be utilizing a single data cache in order to maximize the amount of memory available for loading and creating indices. Compared to runtime configuration of a normal default data cache, there are two aspects to consider:

- Due to the speed of bulk load and volume of data being loaded, we will want to avoid a cache stall as much as we can. Accordingly, we will move the wash marker to 50% of the pool size to support earlier scheduling of the I/O to the I/O subsystem.
- Since bulk operations can leverage large buffers, we will increase the large I/O pool to 50% the size of the cache.

As an example configuration, the following configuration assumes a total of 80GB physical memory reserved for ASE, which would leave a total of around 62GB for data caches.

```
use master
go
-- Calculate the amount of remaining memory to re-increase the default data cache
select cast(sc.name as varchar(30)) as name, (scc.value*2/1024) as MB
  from sysconfigures sc, syscurconfigs scc
 where sc.name like 'total%memory' and sc.config = scc.config
go
exec sp_cacheconfig 'default data cache', '62G'
go
-- Partition default data cache to reduce spinlock contention. The general rule
is:
-- cache partitions >= number of threads in syb_default_pool, but being within
-- 8, 16, 32 or 64):
exec sp_cacheconfig 'default data cache', 'cache_partition=16'
go
-- During import, we use 50% of default data cache for large IO.
exec sp_poolconfig 'default data cache', '31G', '128K'
go
-- Each cache pool has a wash area, which determines the point in the LRU chain
-- when ASE will consider a dirty page for background write to disk. During
import,
-- we set the wash size of each cache pool to 50% of the cache pool size:
exec sp_poolconfig 'default data cache', '16K', 'wash=15G'
exec sp_poolconfig 'default data cache', '128K', 'wash=15G'
-- Configure APF ratio of 50% for any cache in the system
exec sp_configure 'global async prefetch limit', 50
go
```

This can be made more dynamic through a script. See Dynamic Cache Sizing [page 75] in the appendix of this document (don't forget to set *@mode = import* before executing the script!).

As some of the configurations made are not dynamic, you need to reboot ASE before continuing with the data import.

4.5.3 Migration Monitor Settings for Data Import

In addition to the usual settings made in the `import_monitor_cmd.properties` file,

- If using `migmon` and the staged approach, set the `omit` parameter to `PIV`. This will avoid any index creation during the data import phase. If using `distmon`, you can leave the index creations in place
- Set the `loadArgs` parameter to `-c 100000 -loadprocedure fast`. This tells `R3load` to make use of bulk load (BCP) instead of the much slower and fully logged INSERT, as well as to commit every 100000 rows. This 100000 row commit frequency is important to avoid running out of locks while allowing parallel inserts on split packages.
- Set the `orderBy` parameter to `size` to start with the largest packages.
- Set the `jobNum` parameter to the number of physical cores available on the host. This will cause over-assignment of CPU resources due to the additional ASE engines configured on the host, but is best practice for this phase of the import.

Once the `import_monitor_cmd.properties` file has been prepared, start the Migration Monitor and wait until all packages have been imported.

> i Note
>
> The `MIGMON.SAR` archive can be found in the `/COMMON/INSTALL` subdirectory of the Software Provisioning Manager.

4.5.4 Troubleshooting Data Import

SQL Error 1204: ASE has run out of LOCKS

If R3load fails during a table load and you attempt restart it, you may experience problems with ASE running out of locks or the transaction log filling. The problem is caused by the `R3load` recovery mechanism, which deletes all of the rows out of the partially filled table and restarts the load again. Unlike bulk inserts, any delete operation is logged. Consequently, really large tables could result in the transaction log. In addition, ASE could run out of locks for all the deletes on that table.

The recommended solution is to first truncate the table manually (using the `TRUNCATE TABLE` command), and then restart the failed load. Note that in case of a split table all packages of that one table need to be set to status *err* in the corresponding `R3load TSK` files!

SQL Error 1105: Can't allocate space for object ... in database SID because 'logsegment' segment is full/has no free extents

If ASE runs out of log space, you are either running `R3load` without option `–loadprocedure = fast`, or you have facing the problem described for *SQL Error 1204*.

SQL Error 712: There is insufficient heap memory to allocate n bytes

Increase the number of user connections by about 20%, then restart `R3load`.

SQL Error 716: Unable to allocate memory of size 16384 bytes from the 'compression info' pool

Increase the ASE compression info pool size by a factor of two.

SQL Error 2846: SQL function 'dbo.sp_drv_getcomment' not found

Rerun the import monitor.

R3load hangs during bulk copy

There is a known hang issue during `R3load bulk insert` when a high number of parallel import jobs are involved. Increase the number of user connections by about 20%, then restart ASE and `R3load`.

ASE Error Log reports 'Index (objid ..., in did 3) in dbid 4 is not get enough benefit, will never try to compress this index'

This is not an error and can be ignored. If index compression is enabled, ASE will attempt to compress index pages after first 2 extents are allocated. If these first two extents do not show a reasonable compression rate, ASE will automatically disable compression of only this index.

4.5.5 Optimizing the ASE Configuration for Index Creation

Once the data import is done, the next major step is to create all the indices. Creating the indices will require a lot of CPU, I/O bandwidth and memory resources – the more resources granted to ASE, the faster the index creation will complete. Since SAP is not available at this point, and since the R3load processes require minimal resources during index creation, it often makes sense to configure ASE for the maximum memory and CPUs possible and then after creating the indices, scaling it back for SAP Central Instance or other application usage.

Note that we are following one of two different scenarios.

- The first is a local `migmon`-based import in which local `R3load` processes were consuming at least some CPU resources during the raw data import. In this case, the ASE instance can now be increased to reduce the overall create index time.
- The second is a distributed `distmon`-based implementation in which the ASE instance was essentially isolated from the `R3loads`. In this case, the ASE instance should already be configured to consume most of the resources available on the host.

Review and reconfigure both CPU and memory resources for ASE using the steps described in section Optimizing the ASE Configuration for Data Import [page 28] as a template.

4.5.6 Index Creation

As of SAP ASE versions 15.7 SP130 and 16.0 SP1, creation of optimizer statistics during parallel index creation has become stable. At the same time, the use of multiple parallel workers for index creation can be globally limited to a minimum table size. As a result, the best practice is to:

- Use the `statistics hashing` option for all index creations and thus completely eliminate a subsequent update statistics step.
- Use 3-5 consumers/workers for all index creations. Only exception are tables that have been exported sorted (see notes on `DDLSYB.TPL` below).

Defining the Level of Parallelism

SAP ASE supports index creation using multiple parallel threads. These threads are called *worker processes* and are taken from a worker process pool whose size can be configured with parameter *number of worker processes*.

The actual decision on whether parallelism is used for a specific index creation, and to what degree, is made as indicated in the following graph:

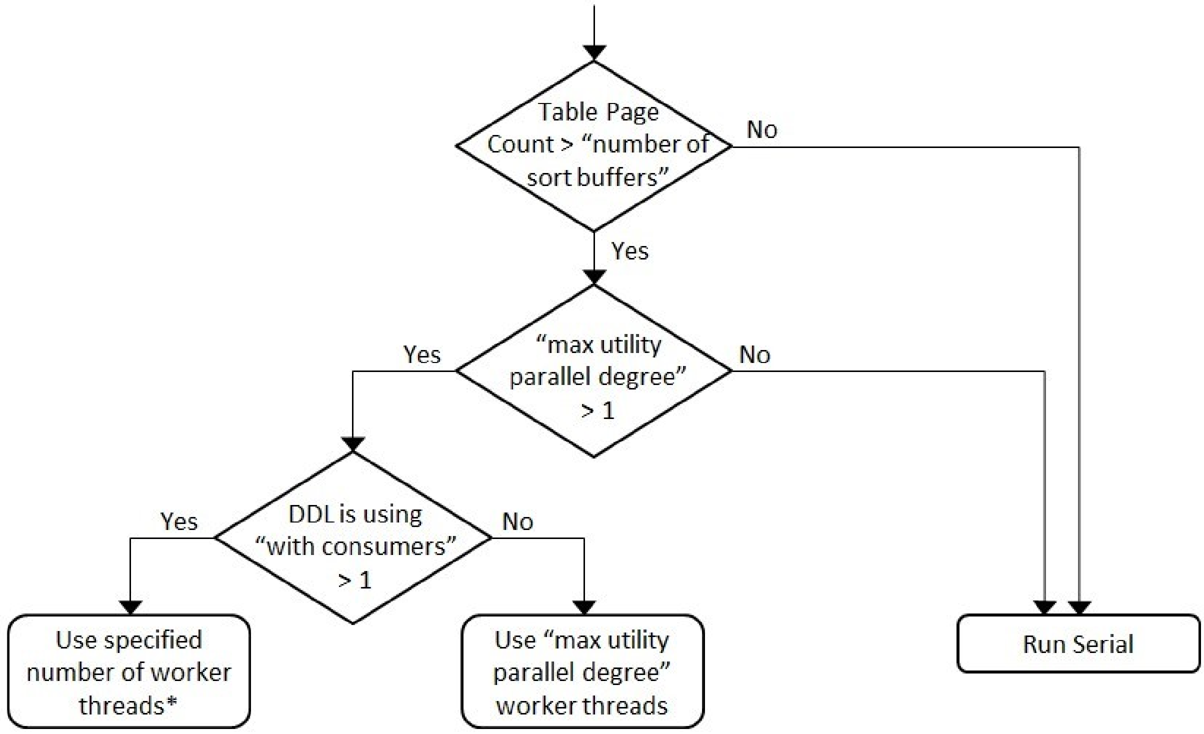


Table Page Count > "number of sort buffers"
No
Yes
"max utility parallel degree" > 1
Yes
No
DDL is using "with consumers" > 1
Yes
No
Use specified number of worker threads*
Use "max utility parallel degree" worker threads
Run Serial

* with max of "max utility parallel degree"

The recommended number of consumers used for a single index creation is 3 to 5, with 5 being the upper limit. To specify the number of consumers used to build an individual index, modify the DDL template files `DDLSYB.TPL` and `DDLSYB_LRG.TPL`. In addition, ensure that the DDL template files make use of the *statistics hashing* option to avoid the separate statistics step.

Note

This does not apply for SAP BW tables, such as fact, PSA or ODSO tables that have been created with report `SMIGR_CREATE_DDL`. The PLL degree has to be specified before the execution of the report. For more information, see the section above: Preparations for SAP BW [page 14] .

Note that tables that have been exported sorted should be mapped to `DDLSYB.TPL`, while all other tables should be mapped to `DDLSYB_LRG.TPL`. As the sorted export does not require re-sorting the data for creation of the primary key, `DDLSYB.TPL` does make use of the *sorted_data* option, instead of specifying the *consumers* option.

```
(DDLSYB.TPL)
prikey: AFTER_LOAD ORDER_BY_PKEY
seckey: AFTER_LOAD
cretab: CREATE TABLE &tab_name&
         ( /{ &fld_name& &fld_desc& /-, /} ) &compression&
drptab: DROP TABLE &tab_name&
crepky: CREATE &unique& INDEX &pri_key&
        ON &tab_name&
         ( /{ &key_fld& /-, /} )
        with sorted_data, statistics hashing
drppky: DROP INDEX &tab_name&.&pri_key&
creind: CREATE &unique& INDEX &ind_name&
        ON &tab_name&
         ( /{ &fld_name& /-, /} )
         with consumers = 3, statistics hashing
```

```
(DDLSYB_LRG.TPL)
prikey: AFTER_LOAD
seckey: AFTER_LOAD
cretab: CREATE TABLE &tab_na
        ( /{ &fld_name& &fld_desc& /-, /} ) &compression&
drptab: DROP TABLE &tab_name&
crepky: CREATE &unique& INDEX &pri_key&
        ON &tab_name&
        ( /{ &key_fld& /-, /} )    with consumers = 3, statistics hashing
drppky: DROP INDEX &tab_name&.&pri_key&
creind: CREATE &unique& INDEX &ind_name&
        ON &tab_name&
        ( /{ &fld_name& /-, /} )    with consumers = 3, statistics hashing
```

4.5.7 Migration Monitor Settings for Parallel Index Creation

As described in *Defining the Level of Parallelism*, and assuming `number of sort buffers` being set to 10000 as recommended, the border for use of parallel threads is a table size of 10000 pages, which equals roughly 155MB. As a result, parallel index creation is automatically reserved for only the medium and large tables, as the `consumers` option gets automatically ignored by ASE for all the small tables. The number of indexes created in parallel is defined by the number of `R3load` jobs configured for the Migration Monitor. The number of indexes that can be created in parallel, with each potentially using threads to do parallel sorts, can be derived from the following formula:

```
                                           Number of ASE Worker Processes
Number of Parallel Index Creations = ------------------------------------
                                     Number of Threads per Index Creation
```

So with 80 worker processes configured in the above example configuration, and a number of 3 threads targeted per index creation, we would set the number of parallel jobs in the Migration Monitor to a maximum of 26.

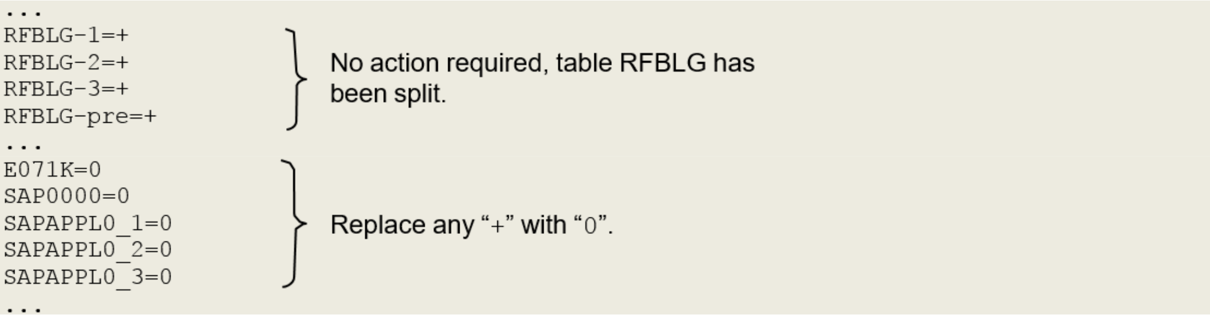
If performing the index creation separately from data import (e.g. `migmon`-based approach), reuse the `import_monitor_cmd.properties` file used for data import, but this time:

- Set the `omit` parameter to `DT`. This will avoid any table creation and data load during the index creation phase.
- Set the `jobNum` parameter to about 75% of the value calculated using the above formula. You may increase the `jobNum` parameter as the index creations have started and as the CPU and I/O resources on the ASE host do allow more. The `import_monitor_cmd.properties` file is rescanned regularly, you do not need to cancel and restart `migmon`.

To ensure all packages are again processed by the Migration Monitor, the `import_state.properties` file must be adjusted as follows:

- For split tables, the Migration Monitor will automatically add a `<package name>-post` step, which takes care of any index creation. No action is required.
- For any other package, the package status must be changed from `+` to `0`.

Example:



```
...
RFBLG-1=+
RFBLG-2=+
RFBLG-3=+
RFBLG-pre=+
...
E071K=0
SAP0000=0
SAPAPPL0_1=0
SAPAPPL0_2=0
SAPAPPL0_3=0
...
```

No action required, table RFBLG has been split.

Replace any “+” with “0”.

Finally, start the Migration Monitor.

To monitor the number of index creations active as well as the number of worker threads used by each of them, use the following SQL:

```
select spid, ( select count(*)
                         from master..sysprocesses sp2
                       where sp2.fid = sp1.spid ) as workers,
            pt.SQLText
       from master..sysprocesses sp1 left outer join master..monProcessSQLText pt
         on sp1.spid = pt.SPID and pt.LineNumber = 1
      where cmd = 'CREATE INDEX'
go
```

4.5.8 Troubleshooting Index Creation

SQL error 1530: Create index with sorted_data was aborted because of a row out of order

If index creation fails due to data being out of order, the DDL template file mapping is wrong for the respective package. Map the failing package to `DDLSYB_LRG.TPL` to omit the `sorted_data` option for the creation of the SAP primary keys. If the `R3load` task files have already been generated, modify the corresponding command file (`<package name> DT.cmd`).

SQL Error 701: There is not enough procedure cache to run this procedure, trigger, or SQL batch

If ASE runs out of procedure cache during index creation and you cannot increase the actual size of procedure cache due to memory constraints, decrease the configuration for `number of sort buffers`.

Index (objid X, indid Y) in dbid 4 is not get enough benefit, will never try to compress this index

Above message may appear in the ASE error during index creations, and should be treated informal. ASE will start compression of index pages after allocating the 3rd extent. If compression of the first two extents does not show a reasonable compression rate, ASE will disable compression of the index.

timeslice error: spid xxx exhausted its 'time slice' of 100 milliseconds and additional 'cpu grace time' of 1000 ticks (100000 milliseconds)

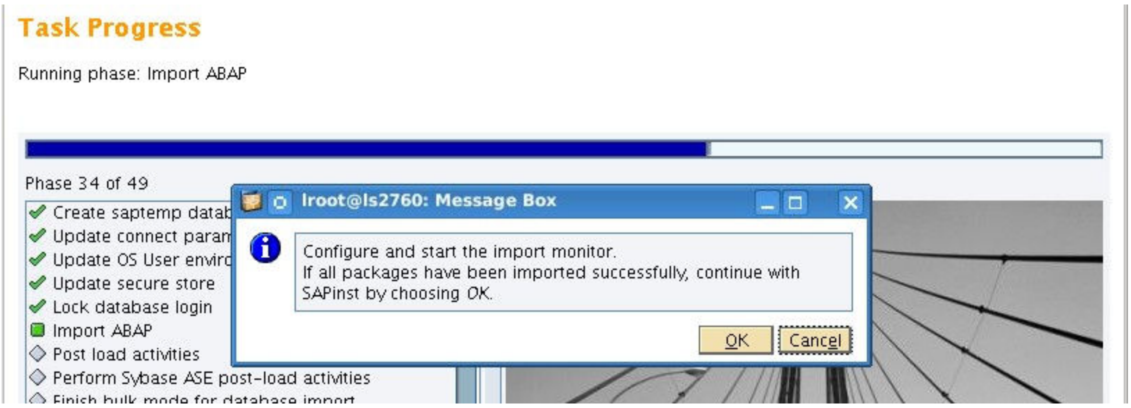
Check the throughput of storage subsystem. If the throughput rates of the storage subsystem are ok, reduce the number of parallel `R3load` jobs by reducing `jobNum` in the `import_monitor_cmd.properties` file. As Migration Monitor does rescan the `import_monitor_cmd.properties` file regularly, you do not need to cancel and restart the import.

The 128K memory pool of named cache default data cache (cache id 0, cachelet id X) is configured too small for current demands (state 1)

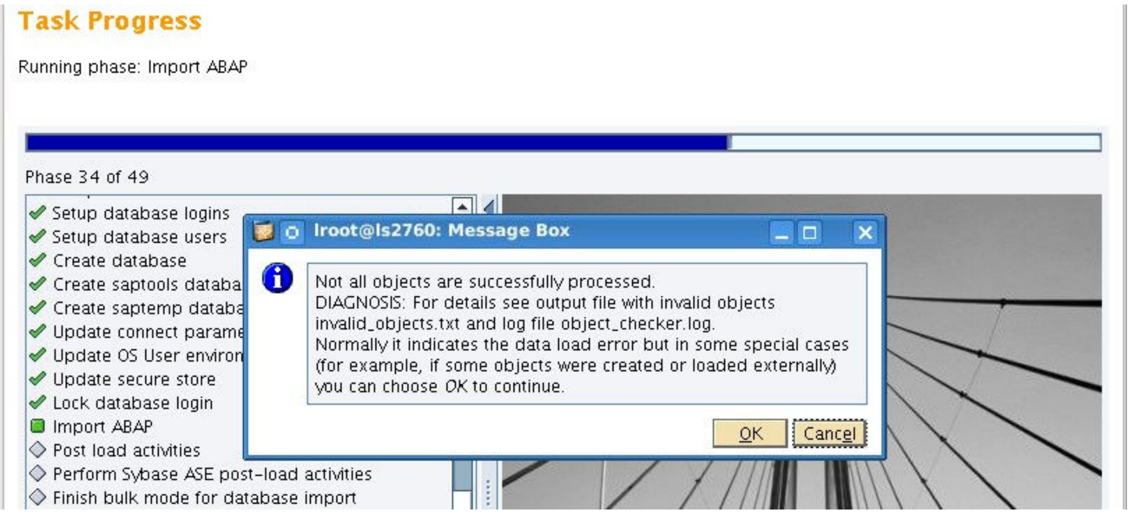
Increase the size of the 128K cache pool at the cost of reducing the 16K cache pool. Following command gives an example on how to resize the 128K cache pool to a total of 45G:

```
exec sp_poolconfig 'default data cache', '45G', '128K'
go
```

4.6 Completing the Installation Process

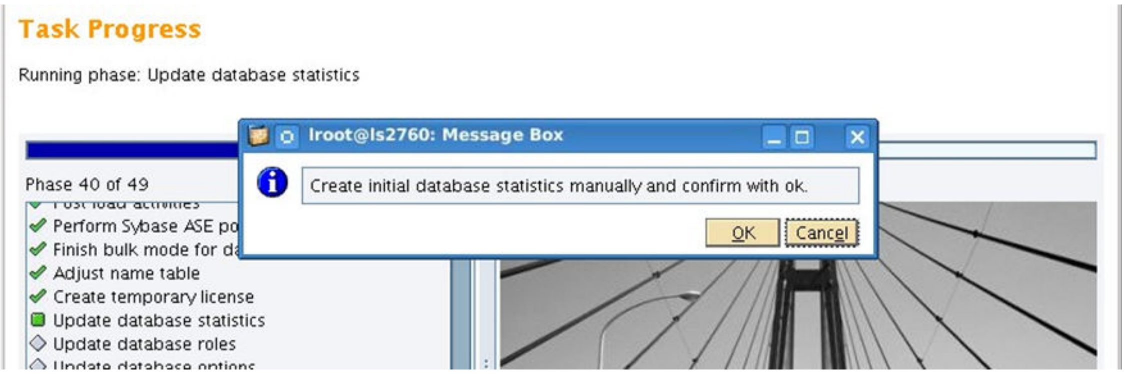
Step	Description
1.	After the Migration Monitor has finished successfully, we can continue with the installation process by choosing `OK`.  Task Progress Running phase: Import ABAP Phase 34 of 49 lroot@ls2760: Message Box Configure and start the import monitor. If all packages have been imported successfully, continue with SAPinst by choosing *OK*. ✓ Create saptemp datab ✓ Update connect param ✓ Update OS User enviro ✓ Update secure store ✓ Lock database login 🟩 Import ABAP ◇ Post load activities ◇ Perform Sybase ASE post-load activities ◇ Finish bulk mode for database import OK Cancel

2.	Since we have processed data import and index creation in two separate runs of Migration Monitor, the installer has trouble identifying whether all import tasks have been executed successfully.
	To ensure everything is ok, do a quick scan for pending or failed import tasks as follows:
	`mydbhost:/<importDir>/DATA # grep "xeq" *.TSK`
	`mydbhost:/<importDir>/DATA # grep "err" *.TSK`

3.	If step 2 returned no pending or failed import task, continue with the installer by again choosing `OK`.  Task Progress Running phase: Import ABAP Phase 34 of 49 lroot@ls2760: Message Box Not all objects are successfully processed. DIAGNOSIS: For details see output file with invalid objects invalid_objects.txt and log file object_checker.log. Normally it indicates the data load error but in some special cases (for example, if some objects were created or loaded externally) you can choose *OK* to continue. ✓ Setup database logins ✓ Setup database users ✓ Create database ✓ Create saptools databa ✓ Create saptemp databa ✓ Update connect parame ✓ Update OS User environ ✓ Update secure store ✓ Lock database login 🟩 Import ABAP ◇ Post load activities ◇ Perform Sybase ASE post-load activities ◇ Finish bulk mode for database import OK Cancel

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>4.</td><td>

After execution of some more installation steps, the installer will stop at step *Update database statistics* and wait for external creation of table statistics.



Task Progress

Running phase: Update database statistics

Phase 40 of 49

✓ Post load activities
✓ Perform Sybase ASE po
✓ Finish bulk mode for d
✓ Adjust name table
✓ Create temporary license
🟩 Update database statistics
◇ Update database roles
◇ Update database options

Iroot@ls2760: Message Box

Create initial database statistics manually and confirm with ok.

Since we have already created table statistics on all application tables during creation of indexes, the only manual step required is to update statistics on the ASE system tables:

```
use <SID>
go
update statistics sysobjects
go
update statistics sysindexes
go
update statistics syscolumns
go
update statistics syspartitions
go
update statistics systabstats
go
update statistics syscomments
go
update statistics sysstatistics
go
```

After that, we simply continue by choosing `OK`.

> **Note**
>
> You can double-check existence of statistics for all indexed columns using the SQL command show in the appendix.

</td></tr>
</table>

4.7 Post-processing for SAP BW

Perform the following post-processing steps:

1. Carefully follow the post-migration instructions given in SAP Note 888210: *NW 7.**: System copy (supplementary note)*. Execute report `RS_BW_POST_MIGRATION` with variant `&POSTMGRDB&` before you begin to work with the SAP BW system.

2. Execute report `RSSYBDBVERSION` and review its output. Any output highlighted in red should be handled as soon as possible. The error message will point you to SAP Notes with information on how to solve the problem.
3. Execute report `SAP_BWTOOLPGM_SYB` to make sure that the database tables belonging to BW objects have been migrated correctly.
4. Adapt the SAP BW process chains for data load operations to SAP BW to match the best practices for SAP ASE:
    1. Delete the step *Delete Indexes* in the process chains.
    For BW systems running on SAP ASE, we recommend loading data using existing indexes of an InfoCube. SAP ASE provides a special performance optimization to support this. Having the indexes in place will also enable BW reporting to run in parallel to data loads. If it is too time-consuming to adapt all process chains, you can make use of a special `RSADMIN` parameter to bypass the *Delete Indexes* step in process chains. See SAP Note 2107943 *Globally deactivate* `DROPINDEX` *process with* `RSADMIN` for details.
    2. SAP recommends that you include a process step *Construct Database Statistics* after the data load operation. Only the statistics of database partitions or tables that have changed significantly will be updated.
5. Check if the Automatic Table Maintenance framework was set up as described in section *Set Up Automatic Table Maintenance [page 44]*. The status of the framework should be checked on a regular basis. It is essential for the performance of SAP BW on SAP ASE.

5 Basic SAP Runtime Configuration

After the SAP system installation (or migration), before you can release the system to runtime users, the following SAP-related configurations should be performed.

5.1 SAP Profile Parameters

There are some database platform-related SAP profile parameters which can negatively influence SAP ASE's performance, and thus should be set as described in SAP Note 1996340.

In addition, consider tuning the number of cached statements used by SAP work processes (`dbs/dbs/syb/cache_size`) to 300 as per SAP Note 1954245. See also Configuring Statement and Procedure Cache [page 50]

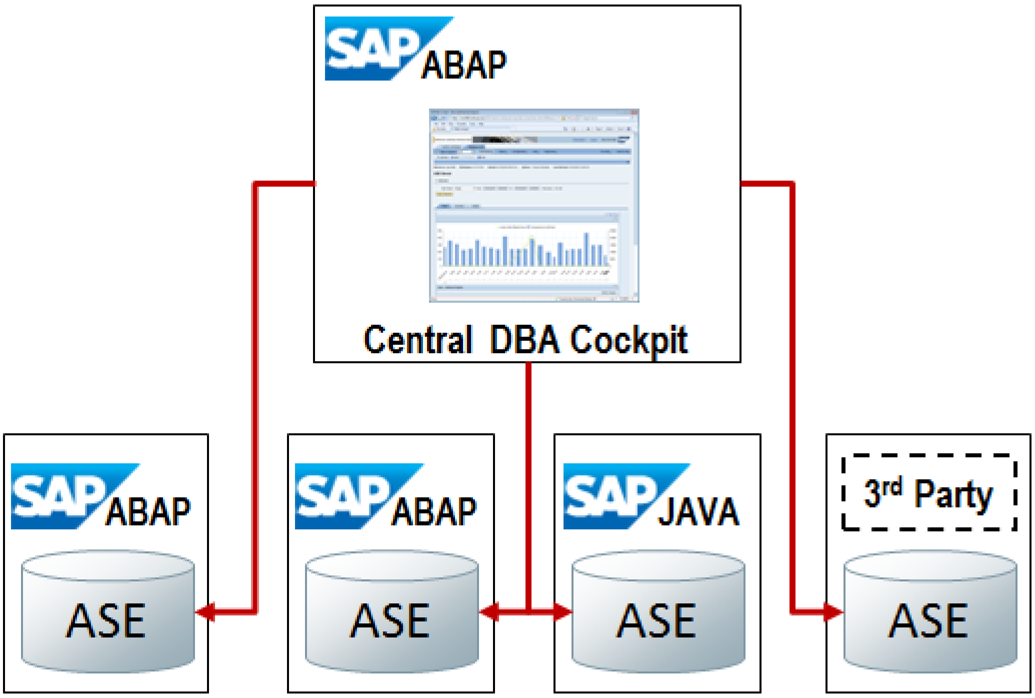
5.2 Configure DBA Cockpit

Enable the WebDynpro services required for DBA Cockpit by following the steps described in SAP Note 1245200.

It is also strongly recommended to apply the DBA Cockpit patch collection according the SAP Basis version of your system:

- SAP Note 1558958: *SYB: DBA Cockpit Correction Collection SAP Basis 7.02 / 7.3*
- SAP Note 1619967: *SYB: DBA Cockpit Correction Collection SAP Basis 7.31*
- SAP Note 1882376: *SYB: DBA Cockpit Correction Collection SAP Basis 7.4*
- SAP Note 2293673: *SYB: DBA Cockpit Correction Collection SAP Basis 7.5*
- SAP Note 2380028: *SYB: DBA Cockpit Correction Collection SAP Basis 7.51*

Consider importing a recent Support Package available for the respective SAP Basis release, which will provide more functionality in the DBA Cockpit. As an alternative, you can use a sandbox system, update it to the latest Support Package available for the respective SAP Basis release, and use it as a central DBA Cockpit for your SAP ASE landscape:



5.2.1 Set Up Automatic Table Maintenance

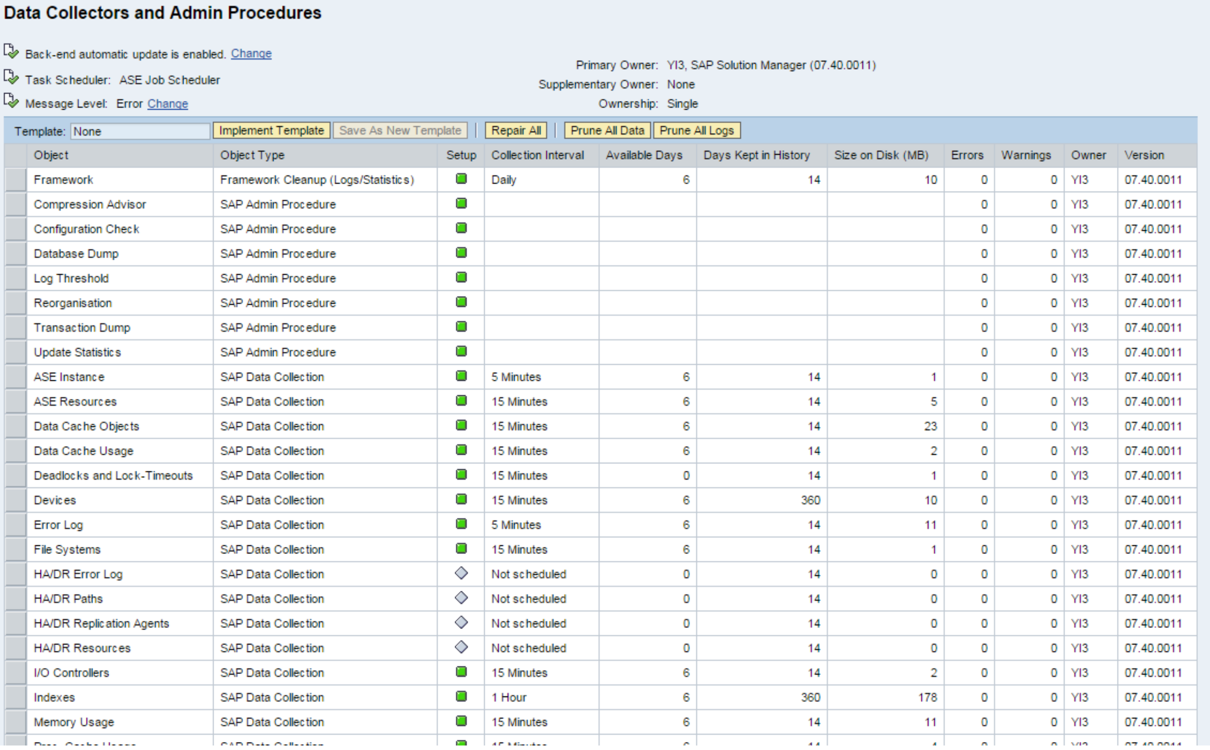
A key aspect of DBA Cockpit is to enable Automatic Table Maintenance. ATM takes care of updating table statistics if the amount of data changed in tables passes a defined threshold. This is important to ensure stable query response times in your SAP system. In addition, ATM evaluates fragmentation of tables and indexes and schedules reorganization if needed.

ATM comes with a default set of maintenance windows and profiles, but is also fully customizable. A more detailed description of ATM can be found in the SAP Community: DBA Cockpit: Automatic Table Maintenance for SAP Adaptive Server Enterprise.

5.2.2 Configure DBA Cockpit Data Collectors

DBA Cockpit for SAP ASE uses a separate database named `saptools` to store DBA Cockpit metadata as well as ASE performance and space metrics. Data collections are scheduled periodically through the ASE built-in job scheduler.

The default data collection configuration schedules metrics collection once an hour and limits the metrics retention to 14 days. This may not be sufficient if wishing to compare end-of-month or end-of-quarter report processing. Also, you may want to have more granular data collection intervals for especially the performance related data collectors (for example once every 5 minutes). Review the default data collection configuration and adjust it according to your needs. In addition, you may want to save the adjusted configuration into a named template for easier transfer to other ASE servers attached to the DBA Cockpit.



Data Collectors and Admin Procedures

Back-end automatic update is enabled. Change

Task Scheduler: ASE Job Scheduler

Message Level: Error Change

Primary Owner: YI3, SAP Solution Manager (07.40.0011)

Supplementary Owner: None

Ownership: Single

Object	Object Type	Setup	Collection Interval	Available Days	Days Kept in History	Size on Disk (MB)	Errors	Warnings	Owner	Version
Framework	Framework Cleanup (Logs/Statistics)	🟩	Daily	6	14	10	0	0	YI3	07.40.0011
Compression Advisor	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Configuration Check	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Database Dump	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Log Threshold	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Reorganisation	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Transaction Dump	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
Update Statistics	SAP Admin Procedure	🟩					0	0	YI3	07.40.0011
ASE Instance	SAP Data Collection	🟩	5 Minutes	6	14	1	0	0	YI3	07.40.0011
ASE Resources	SAP Data Collection	🟩	15 Minutes	6	14	5	0	0	YI3	07.40.0011
Data Cache Objects	SAP Data Collection	🟩	15 Minutes	6	14	23	0	0	YI3	07.40.0011
Data Cache Usage	SAP Data Collection	🟩	15 Minutes	6	14	2	0	0	YI3	07.40.0011
Deadlocks and Lock-Timeouts	SAP Data Collection	🟩	15 Minutes	0	14	1	0	0	YI3	07.40.0011
Devices	SAP Data Collection	🟩	15 Minutes	6	360	10	0	0	YI3	07.40.0011
Error Log	SAP Data Collection	🟩	5 Minutes	6	14	11	0	0	YI3	07.40.0011
File Systems	SAP Data Collection	🟩	15 Minutes	6	14	1	0	0	YI3	07.40.0011
HA/DR Error Log	SAP Data Collection	◇	Not scheduled	0	14	0	0	0	YI3	07.40.0011
HA/DR Paths	SAP Data Collection	◇	Not scheduled	0	14	0	0	0	YI3	07.40.0011
HA/DR Replication Agents	SAP Data Collection	◇	Not scheduled	0	14	0	0	0	YI3	07.40.0011
HA/DR Resources	SAP Data Collection	◇	Not scheduled	0	14	0	0	0	YI3	07.40.0011
I/O Controllers	SAP Data Collection	🟩	15 Minutes	6	14	2	0	0	YI3	07.40.0011
Indexes	SAP Data Collection	🟩	1 Hour	6	360	178	0	0	YI3	07.40.0011
Memory Usage	SAP Data Collection	🟩	15 Minutes	6	14	11	0	0	YI3	07.40.0011

The default `saptools` database size is 2 GB data and 200MB log. If increasing granularity of data collection or extending the retention periods, you need to increase the `saptools` database size. For medium or large setups, SAP recommends to initialy resize the `saptools` database to 10GB data and 1GB log.

6 Basic ASE Runtime Configuration

After the SAP system installation, before you can release the system to runtime users, the following ASE-related configurations should be performed.

6.1 Mandatory Settings

Some of the mandatory or recommended configuration settings are not set by the installer. Make sure SAP ASE is configured according to SAP Note 1539124 before you continue. Specifically, you should check the following SAP Notes depending on the SAP ASE version you are using:

- SAP ASE 15.7: SAP Note 1749935
- SAP ASE 16.0: SAP Note 1581695

In these notes there are `Recommended [REC])` and `Required ([REQ])` configuration parameters. Double check the required `[REQ]` configuration parameters to ensure that none are set to an unsupported value.

> Note
>
> Recent versions of the DBA Cockpit for SAP ASE include an automated check of these parameters. See SAP Note 1956005 for details.

6.2 ASE Sizing

During setup time, some of the memory and CPU resources were configured to be more optimal for the data load and index creation processes. These settings need to be reviewed and optimized for a production runtime environment.

Following are examples of system sizes and recommended ASE configurations for SAP Business Suite systems:

Configuration	Tiny	Small	Medium	Large	Extra Large
Rough DB Size	<512 GB	<1 TB	1-3TB	~5TB	10TB+
ASE Engines / User Threads	2	2-4	4-10	12-18	24+

Configuration	Tiny	Small	Medium	Large	Extra Large
ASE Total Memory	8-16GB	16-32GB	64-128GB	256-384GB	512GB
Locks	1,000,000	1,500,000	5,000,000	10,000,000	15,000,000+
Procedure Cache	1GB	2GB	4GB	8GB	16GB
Statement Cache	512MB	0.5-1GB	1-2GB	2-4GB	5GB
Log Cache	256MB	512MB	1GB	1-2GB	2-4GB
System Tables Cache	64MB	64MB	128MB	256MB	512MB
Queue Cache	(none)	256MB	512MB	1GB	2GB
Temp. DB Cache	(none)	512MB	1GB	1GB	2GB
saptempdb size	32GB	64GB	64-128GB	150GB	200GB
saptools size	5GB	10GB	10GB	20GB	20GB

The following table shows examples of system sizes and recommended ASE configurations for SAP BW:

Configuration	Tiny	Small	Medium	Large
Rough DB Size	<512 GB	<1 TB	1-3TB	~5TB
ASE Engines / User Threads	4-8	10-16	12-24	24+
ASE Total Memory	16-32GB	32-128GB	128-256GB	256-512GB
Locks	2,500,000	5,000,000	7,500.500	10,000,000
Procedure Cache	1-2GB	2-4GB	4-8GB	8-16GB
Statement Cache	512MB	512MB	0.5-1GB	1-2GB
Log Cache	256MB	512MB	1GB	1-2GB
System Tables Cache	64MB	64MB	128MB	256MB
Queue Cache	(none)	256MB	512MB	1GB
Temp. DB Cache	(none)	1GB	2GB	4GB
saptempdb size	32GB	64GB	128GB	150GB
saptools size	5GB	10GB	10GB	20GB

While the values in the table above are a good starting point, all memory pools need to be carefully monitored and adjusted to the specific workload patterns of your system.

Focus areas for SAP Business Suite systems are:

- Statement cache size, which may need additional tuning depending on the average number of active SAP work processes and depending on the number and types of Business Suite scenarios operated on the system.
- Number of locks, which may need additional tuning depending on the lock consumption caused by Z-programs. Also Z-programs possibly need to be adjusted to reduce the number of locks acquired within a single database transaction.

Focus areas for SAP BW systems are:

- Number of user threads / ASE engines, which may need additional tuning depending on the number of concurrent BW users and the size of the largest tables (option for increased parallelism of scans).
- Number of locks, which may need additional tuning if data loads involve large packages.

6.2.1 Shrinking ASE Memory

Additional memory may have been granted to ASE during database migration to speed up the data load and index creation process, while during runtime not all of this memory is planned to be used by ASE. In addition, you may wish to have some spare memory for minor cache adjustments during production, or are planning configure other named caches as described in section Advanced ASE Runtime Configuration [page 59]

In either case, you will need to shrink ASE's memory footprint initially. While reconfiguring available memory is dynamic, shrinking memory pools almost always requires a reboot of the ASE server.

Let us assume that a total of 110GB being granted to ASE during migration, while only 80GB being planned to be used for runtime. The main memory consumers in an ASE server are default data cache and procedure cache. Thus we shrink these memory pools along with the ASE total memory before rebooting the ASE server and reconfiguring these as well as other memory pools.

```
use master
go
-- Calculate the amount of memory allocated and used by ASE
select cast(sc.name as varchar(30)) as name, (scc.value*2/1024) as MB from
sysconfigures sc, syscurconfigs scc
where sc.name like 'total%memory' and sc.config = scc.config
go
name                           MB
------------------------------ -----------
total logical memory            109284
total physical memory           112568
-- Reduce procedure cache to about half of its current size
-- (to be reconfigured later on)
exec sp_configure 'procedure cache size', 0, '5G' go
-- Reduce default data cache to about half of its current size, and adjust
-- large IO pool and wash marker settings as a prerequisite
-- (to be reconfigured later on)
exec sp_poolconfig 'default data cache', '16K', 'wash=5G'
exec sp_poolconfig 'default data cache', '128K', 'wash=5G'
exec sp_poolconfig 'default data cache', '20G', '128K'
exec sp_cacheconfig 'default data cache', '40G'
go
-- As above reconfiguration is not dynamic, we are not allowed to reduce the
-- 'max memory' parameter while ASE is still online
```

```
shutdown
go
```

After the ASE server has shutdown, find the ASE configuration file `<SID>.cfg` in the ASE installation directory. Edit the file and search for the following section:

```
[Physical Memory]
  max memory = 57671680
  enable HugePages = DEFAULT
```

If 80GB is the target memory to be used by ASE, adjust the max memory parameter to 80GB * 1024 = 81920 MB * 512 = 41943040 2KB ASE memory pages. Save the changes made and restart ASE before continuing with the following sections.

6.2.2 Configuring Statement and Procedure Cache

Properly tuning procedure and statement caches is extremely critical for SAP systems. If either is undersized appreciably, it could result in significant CPU spikes or saturation as a result of procedure cache spinlock contention, resulting in periods of debilitating performance degradation.

During regular operation of a SAP system, the number of concurrent maintenance operations such as statistics updates or index creations will be little. SAP however utilizes the statement cache extensively, making the procedure and statement cache size one of the more critical configuration components.

An important point is that the prepared statement cache local to each SAP work process may need to be adjusted as per SAP Note 1954245. If using an ASE database shared library (DBSL) older than listed in SAP Note 1909273, the DBSL will use of a default of 2000 statements cached per work process, so reducing this cache to 300 is key.

Assuming a total of 80GB physical memory reserved for ASE, we would start with a 4GB procedure cache and a 1GB statement cache according the *Medium* t-shirt size for SAP Business Suite systems:

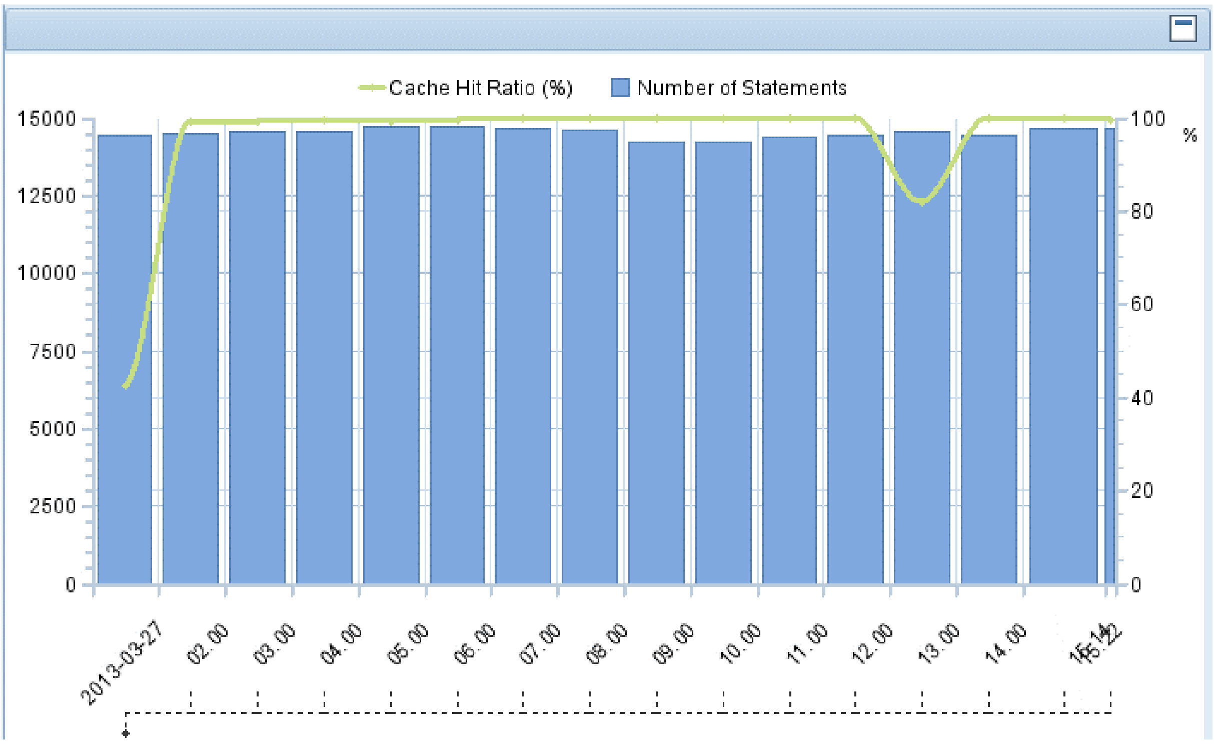
```
-- Procedure cache is the main working memory of ASE. It is used for query
-- compilation, small sorts and work tables, LOB locators, hash joins and it also
-- holds all cached query plans.
exec sp_configure 'procedure cache size', 0, '4G'
go
-- Statement cache is a portion of procedure cache reserved for cached query
plans.
-- It must be at least 256MB in size and should be increased if the number of
query
-- recompiles caused by plan flushes is high.
exec sp_configure 'statement cache size', 0, '1G'
go
-- Sort buffers are used for in-memory sorting. Limit the number of sort buffers
-- available for a single sort to around 2500. This corresponds to a maximum of
-- 256MB a single sort can consume from data cache for in-memory sorting. If your
-- system workload involves very large sort operations, increasing this number
can
-- help improving sort performance.
exec sp_configure 'number of sort buffers', 2500
go
```

Note that the total amount of memory used for procedure cache will be approximately 6GB still - however, 1GB of the procedure cache will be dedicated for the statement cache.

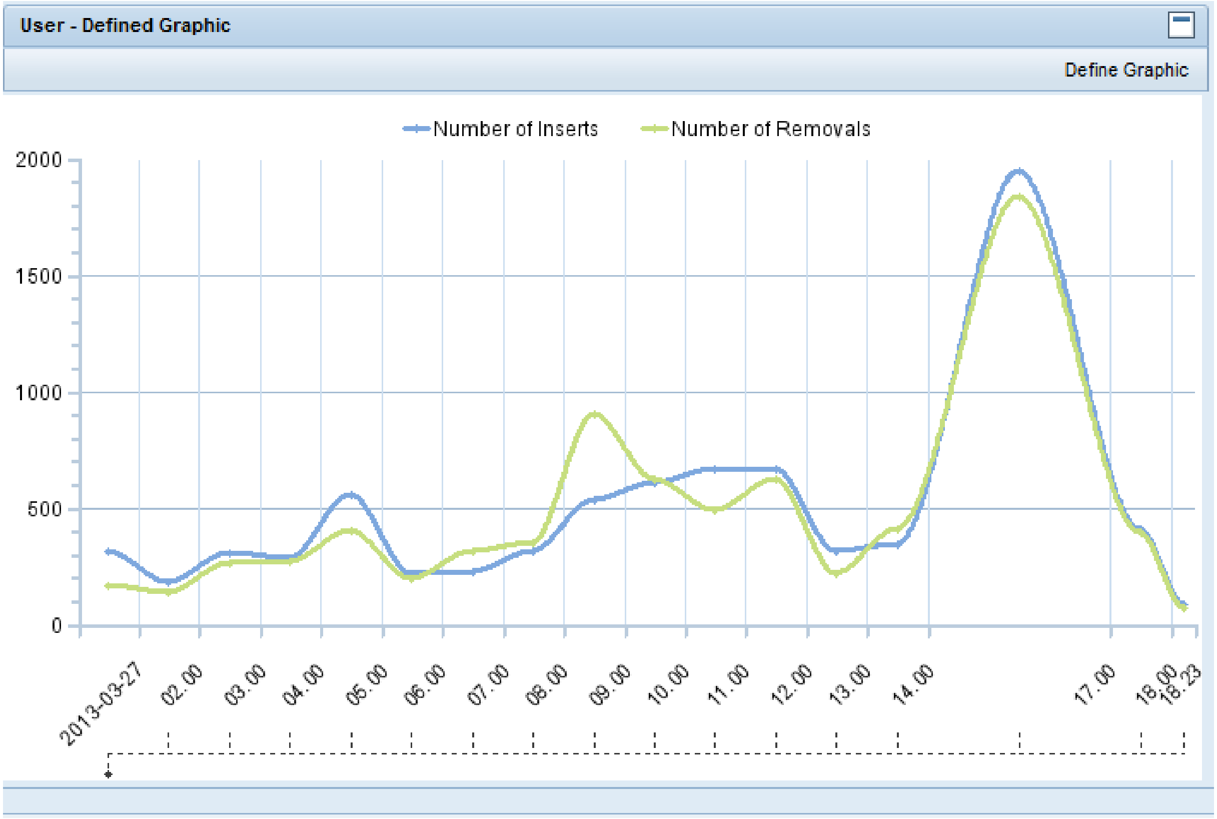
A typical system will have a 4:1 ratio of procedure cache to statement cache at a minimum. A lower ratio such as 2:1 for procedure cache to statement cache may benefit in some use cases, but higher ratios above 4:1 are likely to cause problems.

Also, note that the above are starting configurations. The actual statement cache size necessary will depend on a number of factors such as the number of concurrent users and the transaction rate in the system.

Once in production, monitor the statement cache using DBA Cockpit. Navigate to *Performance* > *Memory Pools* > *Statement Cache Usage*:



Ideally, you do not want the Cache Hit Ratio to be less than 90% in average, but an even better metric is to make sure that the statement turnover isn't exceeding 10% of the total number of statements. This is difficult to see in the above default graph provided by DBA Cockpit. You can however create a user defined graph such as shown below:



Using the earlier graph, we note that there are approximately 14000 statements total in the cache. In the above graph we can see that at one point 2000 of the statements were replaced. If it was a one-time occurrence, this would be no need for further tuning. However, if it happens frequently, since 2000 is much higher than 10% of the total of 14000, we would want to increase the statement cache size. Given that 2000 is roughly 15% of the statements, we would start by increasing the statement cache by about 15% and continue monitoring.

6.2.3 Configuring User Connections

During the load phase, likely the `number of user connections` were set just high enough to support the concurrent load streams. Prior to runtime, the `number of user connections` needs to be set to 4 times the number of SAP work processes across all the SAP application server instances that will be connecting to ASE.

```
use master
go
-- Many SAP applications open temporary secondary database connections to issue
-- a second database transaction independent of the main business transaction
(e.g.
-- number range buffer). Also, the DBSL for SAP Sybase ASE opens two ASE
-- connections for each SAP application connection (different isolation levels).
-- You need to reserve sufficient user connections in ASE.
exec sp_configure 'number of user connections', [total number of SAP WPs * 4]
go
```

One consideration is preventing user connections while the database is in recovery. There are two methods for doing this. The first is to disable logins (except `sa_role` role logins) via the configuration:

```
use master
go
exec sp_configure 'enable logins during recovery', 0
go
```

The alternative and recommended method is to use the configurable database recovery order and hard bindings to `saptempdb` to prevent logins until the SID database is online - and force `saptempdb` to be the last database online using a script similar to:

```
use master
go
-- Enforce a database recovery order that ensures the hard binding prohibits user
-- logins until all the databases are recovered.
exec sp_dbrecovery_order 'saptools', 1, null, strict
exec sp_dbrecovery_order '<SID>', 2, null, strict
exec sp_dbrecovery_order 'saptempdb', 3, null, strict
go
```

6.2.4 Configuring ASE Threads and Parallel Processing

Max Online Engines

The `max online engines` configuration parameter controls the maximum number of ASE query engines in the default and any user defined thread pools that can be brought online. It is recommended that the `max online engines` be set to no higher than the number of physical cores.

Specific tuning recommendations are available in the joint IBM & SAP whitepaper on tuning ASE for AIX. By setting *max online engines* to the maximum supportable by the hardware, this will allow you to dynamically increase or decrease engines as necessary.

Thread Pool Sizing

ASE works best with fewer QP engines running at a higher utilization than with a large number of mostly idle engines. As a result, while it might be tempting to simply configure ASE for the maximum engines, this actually can result in much worse performance than a smaller configuration.

Remember, that in addition to the ASE query engines, there are a number of other threads performing network tasks and disk tasks for ASE that can leverage the threads not used for query engines. In addition, the worker processes for the SAP Central Instance will need CPU cycles - in fact, most of the time, SAP worker processes will need more CPU resources on the DBMS host than the ASE instance itself.

Experience has shown that configuring ASE to only 30-40% of the maximum is likely a good starting point for most SAP systems having a co-located SAP Central Instance.

Thread Pool Idle Timeout

The thread pool idle timeout controls how long an ASE engine is idle before it yields the CPU to the OS. Once yielded to the OS, it will take longer for ASE to wake up the engine to respond to network activity such as new queries arriving, so a longer idle timeout increases the responsiveness of the system. However, if the engines are fairly idle, there will be extensive contention on the run queue spinlock as the engines spin looking for work. Consequently, while running with a longer idle timeout may be best from a response time perspective, it might be better to use a shorter timeout to avoid issues with largely idle engines - especially on larger hosts - and especially on development/test systems in which functional tests may not have high concurrency.

Example Configurations

In our earlier example, we discussed a 32 core host. In a runtime configuration, we may be hosting a Central Instance on this host and wish to limit ASE to 16 engines. This can be achieved using a script similar to:

```
use master
go
alter thread pool syb_default_pool with thread count = 16, idle timeout = 250
go
-- increase number of IO threads on Linux/UNIX
if( @@version not like '%Windows%' )
begin
   exec sp_configure 'number of network tasks', 4
   exec sp_configure 'number of disk tasks', 2
end
go
```

In addition, during the load process, we were using parallel processing to speed up index creation. For SAP ERP (or non-BW) systems, you will likely want to disable parallel processing for queries while allowing parallel worker processes for maintenance activities.

```
use master
go
-- disable parallel processing for queries
exec sp_configure 'max scan parallel degree', 1
go
-- limit parallel degree of maintenance activities to 10
exec sp_configure 'max utility parallel degree', 10
go
-- resize the worker process pool to support about 2-3 maintenance activities
-- making use of worker threads
exec sp_configure 'number of worker processes', 30
go
-- set max parallel degree to max of scan and utility parallel degree
exec sp_configure 'max parallel degree', 10
go
```

SAP ASE 16.0 SP01 and higher allows parallel processing for SAP BW queries. While SAP BW benchmarks have shown that SAP ASE scales very well up to a parallel degree of 10, SAP recommends to start with a parallel degree of 4 (=default) and increase it as needed.

The SQL code generation for BW queries adds an AP hint with the desired parallel degree at the end of the statement. You will see hints like this in ST05 SQL traces, the statement cache, or if you display the SQL Query in the Query Monitor (/nRSRT):

```
plan ‘…(parallel 4) (prop … (parallel 4))…’
```

ASE decides at execution time if the parallel degree of the hint is less than or equal to the value of the configuration parameter *max query parallel degree*. If it is higher, SAP ASE will limit the parallel degree to the value in the configuration parameter.

The default value of 4 of the parallel hint can be changed with an `RSADMIN` parameter; use report `SAP_RSADMIN_MAINTAIN` to set a new `VALUE` for `OBJECT= SYBASE_RSDRS_PARALLEL_DEGREE.` The health check report `RSSYBDBVERSION` can be used to display the actual value of `SYBASE_RSDRS_PARALLEL_DEGREE`, if set.

```
use master
go
-- limit parallel degree for plan hints (as BW uses them) to 6
exec sp_configure 'max query parallel degree', 6
go
-- limit parallel degree for hash-based parallel queries to 4 and
-- limit hash-based parallel scans to tables above ~155MB in size
exec sp_configure 'max scan parallel degree', 4
exec sp_configure 'min pages for parallel scan', 10000
go
-- tell ASE to compile serial plan, and make parallel decision and degree at
-- execution time – this gives faster plan compilation times and more robust
-- plan choice
exec sp_configure 'enable deferred parallel', 1
go
-- limit parallel degree of maintenance activities to 10
exec sp_configure 'max utility parallel degree', 10
go
-- resize the worker process pool to support about
2-3 maintenance activities
-- plus a reasonable amount of parallel queries
exec sp_configure 'number of worker processes',100
go
-- set max parallel degree to max of scan and utility parallel degree
exec sp_configure 'max parallel degree', 1
go
```

6.2.5 Configuring the Default Data Cache

During the setup or migration process, the default data cache was configured with an even distribution of memory between the default and large IO pools, and with a wash marker that caused aggressive IO scheduling.

In a production runtime configuration, we need to distinguish between OLAP and OLTP systems:

- OLAP
  For SAP BW systems with OLAP workload, two basic configurations can be used:
  - Configure the whole data cache with 16K I/O pools. It is assumed that a considerable part of the data resides in the data cache. The configuration is designed for efficient data caching and improves the performance if the system has a large memory and relatively small fact and ODSO tables.
  - Configure 80-90% of the data cache with 128K I/O pools and the remaining part with 16K I/O pools. This configuration with big data cache pools improves the performance when working on large data

and relatively small memory configuration. If you expect a lot of disk reads, this configuration can be 20-30% faster than using 16K pools.

- OLTP
  The typical OLTP configuration consists of 10-20% 128K I/O pools and smaller I/O pools for the remaining part. Such a configuration favors 128K reads on large fact tables and results on high turnover on a 128K data cache with a low caching ratio and a high number of disk reads. On modern systems the ratio between cold cache access (reads from disk) and warm cache access (reads from in-memory data cache) can be significant. Reading from memory can be up to 150 times faster than reading from disk. Configure the data cache as large as possible to achieve good performance in SAP BW systems.
  For SAP Business Suite systems with an OLTP workload, very large index or table scans are not expected and not desired. Thus, we segment the major data cache to allow both single page and large IO scans of usually small to medium size.

In both cases, the wash marker should be at approximately 10% of the pool size to cause less aggressive IO scheduling. As an example for SAP Business Suite systems with OLTP configuration, the following configuration assumes a total of 80GB physical memory reserved for ASE, which would leave a total of around 62GB for data caches.

```
use master
go
-- Calculate the amount of remaining memory to re-increase the default data cache
select cast(sc.name as varchar(30)) as name, (scc.value*2/1024) as MB
   from sysconfigures sc, syscurconfigs scc
  where sc.name like 'total%memory' and sc.config = scc.config
go
-- Increase the default data cache size to its new size
exec sp_cacheconfig 'default data cache', '62G'
go
-- Resize the 128K pool to achieve a 80:20 split
exec sp_poolconfig 'default data cache', '12G', '128K'
go
-- Shrink the wash markers in each pool to about 5-10% of the new pool sizes
exec sp_poolconfig 'default data cache', '16K', 'wash=5G'
exec sp_poolconfig 'default data cache', '128K', 'wash=1200M'
go
-- Partition default data cache to reduce spinlock contention. The general rule
is:
-- cache partitions >= number of threads in syb_default_pool, but being within
-- 8, 16, 32 or 64):
exec sp_cacheconfig 'default data cache', 'cache_partition=16'
go
-- Configure APF ratio of 10% for any cache in the system
exec sp_configure 'global async prefetch limit', 10
go
```

Same example for an SAP BW system with OLAP workload:

```
use master
go
-- Calculate the amount of remaining memory to re-increase the default data cache
select cast(sc.name as varchar(30)) as name, (scc.value*2/1024) as MB
  from sysconfigures sc, syscurconfigs scc
  where sc.name like 'total%memory' and sc.config = scc.config
go
-- Increase the default data cache size to its new size
exec sp_cacheconfig 'default data cache', '62G'
go
-- Shrink the wash marker of the 16K pool to about 5-10% of the new pool size
exec sp_poolconfig 'default data cache', '16K', 'wash=5G'
go
```

```
-- Partition default data cache to reduce spinlock contention. The general rule
is:
-- cache partitions >= number of threads in syb_default_pool, but being within
-- 8, 16, 32 or 64):
exec sp_cacheconfig 'default data cache', 'cache_partition=16'
go
-- Configure APF ratio of 10% for any cache in the system
exec sp_configure 'global async prefetch limit', 10
go
```

At this point, ASE should print informational messages describing what it was able to accomplish online. If large portions of the ASE memory where already in use, which usually only is the case if the SAP system is already in production, you need to reboot ASE to ensure all changes become active.

This can be made more dynamic through a script. See Dynamic Cache Sizing in the appendix of this document (don't forget to set **`@mode = 'runtime'`** before executing the script!).

6.3 Adding a Loopback Entry to ASE Interfaces

When it boots, the ASE server interrogates the interfaces (`interfaces` file in Linux/UNIX and `sql.ini` file in MS Windows) for the corresponding server entry. Once the server entry is located, ASE will then scan for *master* entries and attempt to start network socket listeners using the hostname, port numbers and network driver listed. If multiple *master* lines are found, multiple listeners are created. By default, installation creates a single *master* and *query* line for the server entry using the hostname similar to the below:

```
SID
   master tcp ether hostname.domain.com 4901
   query tcp ether hostname.domain.com 4901
```

It is recommended that a loopback entry at 127.0.0.1 be added to the file as shown below. Note that the IP address is specified instead of `loopback` or `localhost`, and that the entry is first in the list.

```
SID
   master tcp ether 127.0.0.1 4901
   query tcp ether 127.0.0.1 4901
   master tcp ether hostname.domain.com 4901
   query tcp ether hostname.domain.com 4901
```

Normally, when connecting to SAP ASE, clients that do not use LDAP will use the `interfaces/sql.ini` file to locate the server. It will attempt to establish a connection using the information on the first query line found for the requested server name. If that fails, it will attempt a connection using information for the second (or subsequent) query lines. In attempting the connection, DNS host name resolution may be required. By allowing the server to listen on 127.0.0.1, any local client will be able to connect regardless of whether the DNS service is unavailable or even if the network interface card has failed.

6.4 Modifying OCS.CFG for Third Party Compatibility

The default installation for SAP sets the configuration `net password encryption reqd` to *1*, forcing users to connect using connection properties that ensure the password for the connection is sent encrypted vs. clear text. Unfortunately, some 3rd party tools, such as database backup utilities, may not support this option. Rather than leaving the configuration at *0*, an alternative is to modify the `ocs.cfg`. (`$SYBASE/$SYBASE_OCS/config/ocs.cfg or %SYBASE%\%SYBASE_OCS\ini\ocs.cfg`) for the program name. For example, assume we have a database backup utility that simply calls `isql`, a good `ocs.cfg` would be:

```
; This is the external configuration definition file.
;
[DEFAULT]
    ; This is the default section loaded by applications that use
    ; the external configuration (CS_EXTERNAL_CONFIG) feature, but
    ; which do not specify their own application name (CS_APPNAME).
    ; Initially this section is empty - defaults from all properties
    ; will be the same as earlier releases of Open

[isql]
    CS_SEC_ENCRYPTION = CS_TRUE
    CS_OPT_QUOTED_IDENT = CS_TRUE
```

The result is that not only will password encryption be enabled for 3rd party tools, but also quoted identifiers will be enabled. This may not be important for database backup utilities, but it may be necessary for utilities that operate at the table, column or object level.

7 Advanced ASE Runtime Configuration

The basic ASE runtime configuration is not optimal for larger business critical systems. These systems will need some additional tuning and optimization based on workload. While the exact tuning/optimization is largely site specific, this section contains a number of generic advanced configurations that are likely to be common across most installations.

7.1 Log and Other Named Caches

For performance reasons, it is best to have a few named caches to ensure that critical tables are always retained in memory. For ASE, this includes the transaction log of mission critical databases as well as key system tables. In addition, known application volatile tables such as the SAP queue tables should also have a separate cache to improve I/O performance. See also SAP Note 1724091.

```
use master
go
-- reduce default data cache to create a room for named caches. For example,
-- if we earlier had a 62GB default cache and now want to reduce to 40GB
-- to make room to create other caches (and then re-increase this to fill the
space)
exec sp_cacheconfig 'default data cache', '40G'
go
-- stop and restart ASE
shutdown
go
```

When shrinking a data cache, a reboot of ASE is required before memory get released for creation of other named data caches. In addition to the named data cache for the transaction log, it is best practice to increase log IO size from 16K to 32K as well as ensuring the user log cache configuration is an even multiple of the log IO size.

```
use master
go
set quoted_identifier on
go
-- Create a separate 32K log cache (no cache partitions!),
-- size =    512MB minimum to 1024MB maximum
-- for the 4K pool as a starting configuration (to be increased or decreased as
-- necessary based on post-migration monitoring).
exec sp_cacheconfig 'log_cache', '1280M', 'logonly', 'strict',
'cache_partition=1'
-- Move most of the log cache to the 32K pool…some 16K will be needed for
rollbacks
-- and other large transactions
exec sp_poolconfig 'log_cache', '1024M','32K'
go
-- Configure a matching ULC (ASE 15.7 only - ASE 16.0 will use PLCBLOCKS) if
@@sbssav < '16.0.00.00'
begin
  exec sp_configure 'user log cache size', 32768
end
```

```
go
-- Create the 'queue cache' (at least 256MB)
-- Since large scans are not expected to happen on these tables, it is not
-- necessary to define a 128K IO pool.
exec sp_cacheconfig 'queue_cache', '512M', 'mixed','relaxed','cache_partition=4'
go
-- Create the 'systables cache' (128M to 512M)
exec sp_cacheconfig 'systables_cache', '256M',
'mixed','relaxed','cache_partition=4'
go
```

After creation of named caches is done, resize the default data cache to consume unused memory. See Dynamic Cache Sizing [page 75] (do not forget to set **`@mode = 'runtime'`** before executing the script!).

Finally, the log, system and queue tables need to be bound to their respective caches. In order to do so, the SAP database will need to be put into single user mode.

```
use master
go
print ''
print '<< SAP DATABASE >>'
print ''
go
-- Put the SAP database into single user mode
exec sp_dboption '<SID>', 'single user', true
go
use <SID>
go
if not exists ( select 1 from master..sysdatabases
                    where dbid = db_id() and status & 4096 <> 0 )
begin
  print 'Error: Unable to put <SID> into single user mode'
  select syb_quit()
end
go
checkpoint
go
-- Bind the transaction log to the log cache
exec sp_bindcache 'log_cache', '<SID>', 'syslogs'
exec sp_logiosize '32'
go
-- Bind all system tables to the systables cache
-- (without making use of the same)
select so.name as objname, si.name as indname into #bindcache_objects1
   from <SID>..sysindexes si, <SID>..sysobjects so
  where si.id = so.id and so.type = 'S'
    and so.name not in ('syslogs','sysattributes')
go
declare objects_cursor cursor for
   select objname, indname from #bindcache_objects1 order by objname, indname
go
declare @objname varchar(255)
declare @indname varchar(255)
open objects_cursor
fetch objects_cursor into @objname, @indname
while @@sqlstatus = 0
begin
   if @objname = @indname
   begin
    print '<SID>: binding table %1! to systables_cache', @objname
    exec sp_bindcache 'systables_cache', '<SID>', @objname
  end
  else
  begin
    print '<SID>: binding index %1! to systables_cache', @indname
    exec sp_bindcache 'systables_cache', '<SID>', @objname, @indname
  end
  fetch objects_cursor into @objname, @indname
end
close objects_cursor
deallocate objects_cursor
go
-- Bind the SAP queue tables to the queue cache
declare objects_cursor cursor for
   select so.name as objname, si.name as indname,
          su.name || '.' || so.name as objfqname
    from <SID>..sysindexes si, <SID>..sysobjects so, <SID>..sysusers su
   where si.id = so.id and so.type = 'U'
     and so.uid = su.uid and su.name = 'SAPSR3'
     and so.name IN ('VBDATA','VBHDR','VBMOD','ARFCRSTATE','ARFCSDATA',
                     'ARFCSSTATE','QREFTID','TRFCQDATA','TRFCQIN','TRFCQINS',
                     'TRFCQOUT','TRFCQSTATE','RSBATCHCTRL')
   order by si.id, si.indid
go
declare @objname varchar(255)
declare @indname varchar(255)
declare @objfqname varchar(286)
open objects_cursor
fetch objects_cursor into @objname, @indname, @objfqname
while @@sqlstatus = 0
begin
  if @objname = @indname
  begin
    print '<SID>: binding table %1! to queue_cache', @objname
    exec sp_bindcache 'queue_cache', '<SID>', @objfqname
  end
  else
  begin
    print '<SID>: binding index %1! to queue_cache', @indname
    exec sp_bindcache 'queue_cache', '<SID>', @objfqname, @indname
  end
  fetch objects_cursor into @objname, @indname, @objfqname
end
close objects_cursor
deallocate objects_cursor
go
use master
go
-- Revert single user mode
exec sp_dboption '<SID>', 'single user', false
go
-- Put the saptools database into single user mode
print ''
print '<< SAPTOOLS DATABASE >>'
print ''
go
exec sp_dboption 'saptools', 'single user', true
go
use saptools
go
if not exists ( select 1 from master..sysdatabases
                        where dbid = db_id() and status & 4096 <> 0 )
begin
    print 'Error: Unable to put saptools into single user mode'
    select syb_quit()
end
go
checkpoint
go
-- Bind all system tables to the systables cache
-- (without making use of the same)
select so.name as objname, si.name as indname into #bindcache_objects2
  from saptools..sysindexes si, saptools..sysobjects so
 where si.id = so.id and so.type = 'S'
```

```
   and so.name not in ('syslogs','sysattributes')
go
declare objects_cursor cursor for
  select objname, indname from #bindcache_objects2 order by objname, indname
go
declare @objname varchar(255)
declare @indname varchar(255)
open objects_cursor
fetch objects_cursor into @objname, @indname
while @@sqlstatus = 0
begin
  if @objname = @indname
  begin
    print 'saptools: binding table %1! to systables_cache', @objname
    exec sp_bindcache 'systables_cache', 'saptools', @objname
  end
  else
  begin
    print 'saptools: binding index %1! to systables_cache', @indname
    exec sp_bindcache 'systables_cache', 'saptools', @objname, @indname
  end
  fetch objects_cursor into @objname, @indname
end
close objects_cursor
deallocate objects_cursor
go
use master
go
-- Revert single user mode
exec sp_dboption 'saptools', 'single user', false
go
```

Each of the named caches should be monitored carefully as soon as application testing starts.

- You may decrease the log cache if the number of physical reads is zero, but increase if the number of physical reads and commit times are rather high.
- You may increase or decrease other named caches depending on their hit ratios.

Note

It is not recommended to use a 32K pool for logging in the default data cache. Default data cache is usually partitioned to avoid spinlock contention. This is however problematic for log IO.

Note

SAP applications will only make implicit use of temporary databases (LWPs, work tables during query execution, hash join, sorts, and so on), and all of these activities are minimally logged. In addition, none of the other databases is expected to show any noticeable log volume. For this reason, recommendation is to apply 32K IO logging to only the SAP database.

7.2 Named Cache for SAP Temporary Database

It is also often recommended to have a separate cache for the main temporary database, which in case of SAP is `saptempdb`. Otherwise, table scans and other IO intensive activities in the default data cache could cause worktables and other objects in `saptempdb` to be flushed from memory, causing expensive physical reads to access the worktables. In order to bind a user defined temporary database to a separate named cache, there are a number of conditions that need to be met:

- No users can be bound to that temp. database
- No active sessions using that temp. database

Consequently, the following steps can only be performed when the SAP application is shut down:

```
-- Make sure all SAP processes are shut down and there are no connections as user
-- SAPSR3 or SAPSR3DB before proceeding. Also, you will need to temporarily
unbind
-- the 'sapsa' login from saptempdb.
use master
go
exec sp_tempdb 'unbind', 'lg', 'sapsa', 'db', 'saptempdb'
go
-- disconnect
select syb_quit()
go
-- reconnect
use master
go
-- Verify no active sessions using saptempdb
exec sp_tempdb 'who', 'saptempdb'
go
-- If no active sessions, unbind all the users
exec sp_tempdb 'unbindall_db', 'saptempdb'
go
-- Verify no users bound to saptempdb
exec sp_tempdb 'show'
go
-- Create the tempdb cache
exec sp_cacheconfig 'saptempdb_cache', '1G', 'mixed'
-- Move around 40% of tempdb cache into a large IO pool
exec sp_poolconfig 'saptempdb_cache', '400M', '128K'
go
-- Bind the saptempdb to the cache
exec sp_bindcache 'saptempdb_cache', 'saptempdb'
go
-- If the above is successful, rebind the SAP users
exec sp_tempdb 'bind', 'lg', 'SAPSR3', 'db', 'saptempdb', null, 'hard'
exec sp_tempdb 'bind', 'lg', 'SAPSR3DB', 'db', 'saptempdb', null, 'hard'
go
-- As sapsa is usually used to run update statistics on SAP objects, sapsa is
also
-- bound to the larger saptempdb. However, in case of sapsa, we default to a soft
-- binding so it can log in without saptempdb being available.
exec sp_tempdb 'bind', 'lg', 'sapsa', 'db', 'saptempdb'
go
```

7.3 Optimize Performance of SAP Queue Tables

By default, compression is disabled for volatile SAP tables like the VB- and RFC-queues. Further optimizations are described in SAP Note 1724091.

7.4 Optimize Performance of SAP Lookup and Small Tables

Some of the very small tables might induce heavy decompression overhead due to very frequent read access. Possible optimizations are described in SAP Note 1775764. Note that the term small is relative. Several sites have found benefits in uncompressing all tables less than ~40MB in size compressed. This also decreases spinlock contention on the `CPMEMINFO_SPIN` which controls memory allocations from the heap memory used for data compression/decompression. Candidates for tables that likely should be decompressed include:

- Tables less than ~40MB compressed
- Tables commonly fully in cache that have frequent range scans or index leaf scans

7.5 Optimize On-Row LOB Storage

If not done during migration of the database, you can use the ABAP report attached to SAP Note 1680803 to analyze and optimize the on-row LOB length of large tables.

7.6 Runtime Configuration for SAP BW

7.6.1 Correction Collections

Correction instructions describe the changes that must be made to the source code of programs to correct errors. You can implement the correction instructions that are contained in an SAP Note, but SAP recommends to implement complete and consistent BW correction collections instead of an individual SAP Note. Make sure that you always implement the newest version of a correction collection!

- SAP Note 1605169 *SYB: SAP BW 7.02 Correction Collection*
- SAP Note 1608417 *SYB: SAP BW 7.30 Correction Collection*
- SAP Note 1616726 *SYB: SAP BW 7.31 Correction Collection*
- SAP Note 1821924 *SYB: SAP BW 7.40 Correction Collection*
- SAP Note 2193724 *SYB: SAP BW 7.50 Correction Collection*

- SAP Note 1946164 *SYB: DDIC patch collection for release 7.02*
- SAP Note 1965664 *SYB: DDIC patch collection for release 7.30*
- SAP Note 1965754 *SYB: DDIC patch collection for release 7.31*
- SAP Note 1965755 *SYB: DDIC patch collection for release 7.40*

- SAP Note 1558958 *SYB: DBA Cockpit Correction Collection SAP Basis 7.02 / 7.30**

- SAP Note 1619967 *SYB: DBA Cockpit Correction Collection SAP Basis 7.31**
- SAP Note 1882376 *SYB: DBA Cockpit Correction Collection SAP Basis 7.40*
- SAP Note 2293673 *SYB: DBA Cockpit Correction Collection SAP Basis 7.50*
- SAP Note 2380028 *SYB: DBA Cockpit Correction Collection SAP Basis 7.51*
- SAP Note 1946165 *SYB: Solution Manager Technical Monitoring Patch Collection*

7.6.2 Table Partitioning of Fact Tables of an InfoCube

The physical partitioning of database tables and the handling of SAP BW fact tables have different aspects that have to be taken into account:

- On the one hand table partitioning accelerates query performance by allowing partition pruning. As of SAP ASE16.0 parallel queries are allowed. SAP BW requests can easily be deleted by dropping a partition during the BW collapse operation. Expensive database commands like the update of statistics or the maintenance of indexes will benefit from the smaller amount of data associated to a database partition.
- On the other hand there will be much more overhead in the database optimizer to find the best access plan if there are too many partitions. Critical database limits can be violated with regard to the number of partitions.

For BW on SAP ASE it is always recommended to partition the E fact table of an InfoCube by a time criterion (`0CALMONTH, 0CALYEAR`, or `0FISCPER`). This allows partition elimination (partition pruning) for SAP BW queries. Partitioned E fact tables do not have any negative side effect. Features to merge database partitions for an archived period of time or to add new partitions on top for the coming years are available (*repartitioning*).

The F fact tables of InfoCubes are partitioned by SAP BW requests; each new data load to the InfoCube will create another partition. It is strongly recommended that you perform the BW collapse operation regularly to keep the number of requests, that means the number of database partitions, small. Too many requests will have a negative impact on the performance and can lead to database issues.

The following maximum number of partitions per F table (=number of uncompressed SAP BW requests) are recommended:

- SAP ASE 15.7:
  The maximum of 1000 partitions should not be exceeded to keep query performance acceptable and to avoid database issues.
- SAP ASE 16.0 and higher:
  Make sure that the number of partitions is not much higher than 1000. A higher number of partitions will have a negative impact on performance. A maximum of up to 4000 partitions can be handled by SAP ASE.

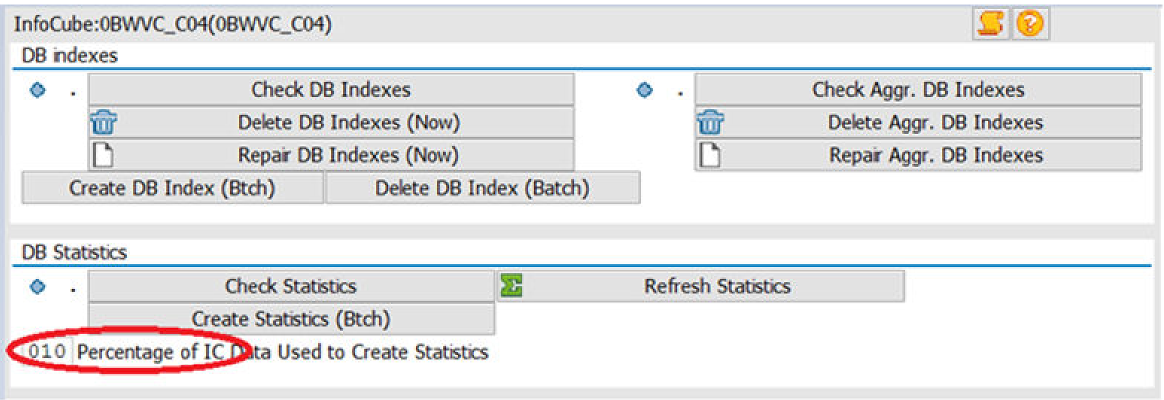
For more information, see SAP Note 2187579 *SYB: Physical partitioning of fact tables*.

SAP Note 1691300 *SYB: Unpartitioned F fact tables for InfoCubes* enables you to avoid the partitioning of F fact tables of dedicated InfoCubes on demand if the cube cannot be compressed frequently.

7.6.3 Table Statistics

Accurate table statistics are essential for the SAP ASE optimizer to find an efficient query plan. Missing or outdated statistics can lead to an inefficient and slow query plan. Very complex SAP BW queries and the large amount of data in SAP BW fact tables tighten this problem.

1. Automatic Table Maintenance (ATM)
   Considering the thousands of tables and indexes a typical SAP system creates on the database, the ATM framework automates manual administrative activities and provides the basis for the query performance of SAP BW. Carefully set up the ATM framework and check if it is working properly on a regular basis!
   For more information, see Set Up Automatic Table Maintenance [page 45]
2. Process Step *Construct Database Statistics*
   The ATM framework is very helpful for tables that grow consistently. The regular maintenance frequency is possibly not appropriate for fast changing tables of the BW star schema. It is not unusual for SAP BW that a significant amount of data is inserted or dropped. The delay between the data load and the update of statistics with ATM can be too long; this will have an impact on BW queries in the meantime.
   Check if your SAP BW process chains include the step *Construct Database Statistics*. In general, SAP strongly recommends to refresh the statistics of an InfoCube and the underlying star schema tables at the end of every SAP BW process chain that loads data to an InfoCube or maintains data of a cube. Perform these steps especially after a database migration!
3. Percentage of data used to create table statistics with sampling
   The `UPDATE STATISTIC` command requires table scans or leaf-level scans of indexes. It can increase I/O contention, use the CPU to perform sorts, and uses data and procedure caches. The use of these resources can adversely affect queries running on the server if `UPDATE STATISTICS` is executed when usage is high. To reduce I/O contention and resources, run the `UPDATE STATISTIC` command for the BW star schema using a sampling method, which can reduce the I/O and time when your maintenance window is small and the data set is large. The percentage to use with sampling for star schema tables depends on your needs. The SAP default value is too small for the needs of SAP BW on SAP ASE. Call the Administrator Workbench InfoCube Management (/nRSA1) and choose the Performance tab.

   010 Percentage of IC Data Used to Create Statistics

   The default value of 10% is much too small. You may test various percentages until you receive a result that fits best. A value greater than 30% is a good starting point. Although a sampling of the data set may not be completely accurate, usually the histograms and density values are reasonable within an acceptable range. The percentage maintained in the InfoCube performance tab is used for updating the statistics of all tables of the InfoCube's star schema: fact tables, dimension tables, SID tables and so on. It can lead to inaccurate statistics if some of these tables are small. In such situations sampling does not make sense. Therefore, a threshold of 10.000.000 rows was introduced with SAP Note 2138611 *SYB: Better table statistics for star schema tables of an InfoCube*. Sampling will only be used for tables with more rows than this threshold.

And sampling will not be used for partitions if a table is partitioned. SAP recommends to implement this SAP Note as soon as possible. The note is part of the Correction Collections (see above). The inaccurate statistics of the star schema tables should be refreshed (deleted and recreated) after the implementation of the SAP Note.

4. Refreshing the table statistics of star schema tables
   You can refresh (*delete and recreate*) the table statistics of the InfoCube's star schema tables if the sampling percentage has been changed, if there are doubts whether the statistics are accurate or not, or if the performance of BW queries is unexpectedly slow. The table statistics of these tables can be refreshed with report `RSSYBREFRESHINFOCUBESTATISTICS` on InfoCube level. You can restrict the statistic refresh to the fact tables, the dimension tables, the SAP BW aggregates if defined, or the tables of the InfoObjects. A test run without refreshing the statistics can be performed if the appropriate flag is set. This can be helpful to identify the affected tables, in case of need. Execute the report as batch job in the background if the statistics of many large tables have to be refreshed. The report `RSSYBREFRESHINFOCUBESTATISTICS` is available with SAP Note 2138611 *SYB: Better table statistics for star schema tables of an InfoCube*.
   Important: Implement SAP Note 2159214 *SYB: ASE Error SQL226* in `RSSYBREFRESHINFOCUBESTATISTICS` in advance if you want to refresh the table statistics.
5. SAP BW Reporting
   Intra-query parallelism for BW Reporting
   SAP BW makes use of ASE's parallel query with SAP ASE 16.0 SP01 and higher. Parallel queries are not supported for SAP BW on SAP Sybase ASE 15.7. The following SAP Notes have to be implemented as a prerequisite in addition; they are part of the BW Correction Collection Notes mentioned above. It is always recommended to implement consistent Correction Collections instead of isolated SAP Notes.
   - SAP Note 2103636 *SYB: Adaptions for SAP ASE 16.0 SP01 PL01*
   - SAP Note 2155343 *SYB: New ASE optimization goal sap_olap*

   It is strongly recommended to collapse the BW InfoCubes frequently to move a large part of the data into E fact tables which should be partitioned by a time criterion. This will enable the ASE optimizer to prune partitions which are not in the selection and to execute the query in parallel.
   The generated queries of the BW reporting will be executed with a parallel degree of 4. This default can be changed with the help of the `RSADMIN` parameter `SYBASE_RSDRS_PARALLEL_DEGREE`. Use report `SAP_RSADMIN_MAINTAIN` to change it; a parallel degree of 0 or 1 means serial execution which is only needed in support situations. The ASE database configuration supports the parallel query execution, see section *Configuring ASE Threads and Parallel Processing [page 53]*.
   - Reducing the complexity of SAP BW queries
     Check SAP Note 514907 *Processing complex queries* especially after a migration to SAP ASE. The threshold value for SAP BW queries should be less than or equal to 50, the threshold value for DataMarts should be less than or equal to 20.
     Use report `RSSYBDBVERSION` to define the values.
   - SAP ASE optimization goal *sap_olap*
     The user defined optimization goal named *sap_olap* was defined and is used for BW reporting queries. The content of this goal can change with different releases of SAP ASE. It is maintained by the ASE upgrade scripts.
   - Fact table hint
     The SAP BW SQL query generator provides optimizer hints for faster query execution.
     The hint named `fact_table` indicates that the database optimizer has to use the snowflake schema optimization strategy. This hint is used in queries with cube fact tables and with queries using ODSO tables.
6. SAP BW Warehouse Management ("WHM")

- Loading data to InfoCubes and database indexes There are two different approaches how to deal with database indexes and data load into InfoCubes.
    - For *some* database systems it is preferred to drop the database indexes of the fact table before loading data into an InfoCube. They have to be created again afterwards.
    - For *other* databases and especially for SAP ASE it is recommended to load data into InfoCubes with existing indexes.

  Keeping the indexes during data load has two major advantages:
    - SAP ASE *offers* a special index maintenance mode (*ins by bulk*) which is useful for BW. The different indexes of the fact tables will be maintained in several concurrent threads at the same time. This concurrent maintenance is faster than dropping the indexes before the load, loading the data into the table without indexes, and recreating them afterwards.
    - To *perform* BW reporting at the same time as the data load operation will benefit from having indexes. It is recommended to remove the process step Delete Indexes of an InfoCube out of BW process chains used for loading data into cubes. A workaround is described in SAP Note 2107943 *P34: PC: Globally deactivate DROPINDEX process with RSADMIN*. Set the RSADMIN parameter, described in the SAP Note to *X*. In this way, the drop index step will be skipped in process chains. This will help customers to change all process chains at the same time. In general, SAP recommends to adapt process chains manually!
- SAP BW Compression / Collapse / Condense
  It is essential to perform the SAP BW compression on a regular basis to keep the number of database partitions small, see section Table Partitioning of Fact Tables of an InfoCube [page 65]. The following SCN document will provide a good overview on compression, in addition to the information provided above: All about Compression
- SAP BW threshold value for data loading
  Use transaction code `RSCUSTV6` *BW Threshold Value for Data Loading* to check and set the following values:
    - *Package Size*
      The value 50,000 is a good starting point. Changing this value requires performance tests and possibly adjustments in the ASE configuration (number of locks, log space, and so on).
    - *Partition Size* (Size of a PSA partition)
      Do not assign a value.
- Enable parallelism on application level for data loads
  Loading data into InfoCubes will benefit significantly from parallel execution. Check your Data Transfer Process (DTP) if parallelism is enabled. In the Maintenance screen for DTPs, choose *Goto* → *Settings for Batch Manager*.
  The *Number of Processes* in the *Settings for Parallel Processing* dialog should be a number of 3-5 and higher depending on the resources available during data load time. SAP ASE has to be configured accordingly, see section Configuring ASE Threads and Parallel Processing [page 53]

8 Database Security

Many corporations have to meet SOX, PCI, HIPPA or other government regulatory or industry compliance for their IT systems. As a result, there not only is a need for separation of duties, but also a need to provide traceability for system administrative actions. SAP applications provide a comprehensive security implementation within the application stack. As a result, the key consideration from a database security standpoint is to ensure compliance of users who connect from outside the SAP application framework.

8.1 Database Logins and Roles

The standard ASE features support the separation of duties as well as granular permissions support the ability to limit the activities of automated processes to strictly those permissions necessary to perform the actions. SAP has used some of these roles and permissions to construct separate logins and roles that are necessary for a standard SAP implementation. However, the standard implementation may not be enough to certify SOX or PCI compliance. The following roles and logins have the specific recommendations.

8.1.1 sa login

By default, the ASE login sa has both the ASE sa_role and ASE sso_role assigned. This is necessary as some sa actions require the ability to modify system tables (typically an system security officer action) as well as allow initial system access to assign the proper roles to other users. A strongly recommended practice that SAP has recommended for a long time – and not just for SOX/PCI compliance – is to lock the sa account and only unlock it when necessary. Some times when sa account may need to be unlocked include:

- When applying SAP ASE software patches that require scripts to be run updating system procedures and system schemas.
- When using the optdiag utility to import statistics into table (needs sso_role to allow system table updates and sa_role to run optdiag).

The default SAP installation locks the sa login.

8.1.2 sapsa Login

The SAP sapsa login takes the place of the ASE sa login, but is restricted to just sa_role which is inherited via the SAP user defined role sap_adm. In addition, the login has sybase_ts_role and if SAP Replication Server is being used to provide HA/DR services, it also has replication_role. In addition, the sap_adm role also provides js_admin_role (necessary for creating and scheduling tasks such as ATM), and sap_mon role (which has ASE's mon_role – necessary for collecting monitoring data into saptools and DBA Cockpit).

The problem with this account, of course, is that it would not meet SOX or PCI compliance as there would be no traceability from a user action standpoint. While the account login is necessary during the initial installation, the intent was that access to the system via this account would only be through SAP's DBA Cockpit – which leverages SAP's application layer security infrastructure. This access, unfortunately, precludes locking the sapsa login. However, SAP does have a published methodology for obfuscating the sapsa login password such that the only access would be through DBA Cockpit. As a result, it is recommended that individual DBA logins be created and be granted the sap_adm role as will be discussed below.

8.1.3 sapsso Login

The SAP sapsso login is similar in function to the sapsa login but with respect to the sso_role. Currently, the sapsso login only has the ASE sso_role assigned. Again, similar to the sapsa login, this account should be only accessed via the SAP application layer and separate SSO logins created. This will be critical when SAP certifies with ASE 16 and the Full Database Encryption option as certain aspects such as key recovery and dual control requirements for higher compliance requires a minimum of two sso_role users (or at least one sso_role and one key_custodian_role login).

8.1.4 Named DBA Logins

It is strongly recommended that individual DBAs have their own logins to ensure SOX or PCI compliance. This is also an ASE security best practice. The SSOs will need to establish which method of password authentication will be used among the following:

- *ASE password policy* – The DBA login password are authenticated by passwords stored in the ASE database and password policy rules such as expiration, digit, repeating passwords, etc. are established using ASE supplied functionality.
- *OS password authentication* – The DBA login passwords are authenticated directly with the operating system using the ASE PAM module. Password policies and rules are set in the operating system.
- *LDAP/Windows AD* – The DBA login passwords are authenticated using LDAP or Windows Active Directory using ASE's Directory Services feature. Password policies and rules are set in the appropriate directory service (LDAP or AD).

> i Note
>
> The password authentication mechanism can be set on a per login name basis in SAP ASE. As a result, the sapsa login would continue to use the standard ASE password authentication while DBA and other non-SAP created logins could use different authentication techniques. In addition, ASE supports Kerberos integration, but this technique is likely not applicable for DBA activities as some ASE supplied tools such as optdiag and others may not support it.
>
> Rather than granting the DBAs sa_role, it is recommended that instead the DBA's be granted the same sap_adm role that is used by the sapsa login. This ensures that the same permissions with respect to

mon_role and js_admin_role are also available. In addition, the sybase_ts_role and replication_role may be selectively granted to specific DBAs as necessary. With respect to auditing DBA activities, see the section on Auditing [page 73] later.

Note that automated jobs scheduled outside the DBA Cockpit framework should **not** be given a DBA login. See the later section on Automated Job Logins [page 71].

8.1.5 Named SSO Logins

Similar to the DBA logins, individual SSO logins should each have individual logins and not use the common sapsso account. This is especially critical if using data encryption as a key best practice for key control requires the use of dual control techniques in which no single user has access to the passwords that are used to encrypt the key encryption keys. The SSO logins can use the same password authentication as used for DBA logins – or not – depending on the decision of the group. It is recommended that at least one SSO either retain the sapsso password – or that at least one SSO use ASE password authentication – or an sso rescue account be created in case of LDAP or external authentication mechanism failure, an SSO can login and reset the password authentication to use ASE's internal mechanism, reset passwords or unlock the ASE sa login as necessary.

8.1.6 Automated Job Logins

Most DBA related regular maintenance can be scheduled via DBA Cockpit and the ASE job scheduler. However, some sites may opt to include other automated jobs outside the SAP supplied framework – such as their own database backup schedules, audit data export processes or other similar functions.

It is recommended that these automated jobs be created using individual logins for each job and that the login be granted **only** the necessary permissions to accomplish that job. For example, rather than granting an external database backup utility sa_role, the login should instead only be granted oper_role.

Tools used by DBAs to supplement monitoring should only be granted sap_mon role. If an appropriate role does not exist that provides the necessary permissions and only the desired permissions, it is recommended that a new role be created and only the necessary permissions be granted to it as far as possible. Creating specific roles with limited narrowly-scoped permissions may be more easily achieved by using the ASE granular permission feature as described in SAP Note 2106688. However, many of these requirements can be met using existing capabilities.

This point of only granting the necessary permissions is critical in preventing DBA's, or others who might have access to those accounts from using the automated job account to bypass any traceability or auditing of actions. For example, consider this slightly modified sample from the ASE documentation (SAP ASE Security Administration Guide):

```
use sybsecurity
go
exec sp_adduser audit_extractor
go
grant select any audit table to audit_extractor
go
```

The result is that an audit extraction automated process can log in and extract audit data periodically and only extract audit information – the audit exaction process does not need an SSO role which could expose the system to users bypassing traceability controls.

8.2 Operating System ASE SID Account

It is a security best practice to make sure that individual users have individual OS accounts as well. When necessary, DBAs and SSOs can use `sudo` or `sesu` to gain access to the SAP software installation to restart ASE or companion processes such as ASE Backup Server.

It is important to realize that such access should not be viewed as a security mechanism for controlling access to the ASE server itself by restricting access to the client software. While the standard ASE installation includes client software, numerous other packages also include ASE client components – including ODBC and JDBC libraries that do not even need ASE OpenClient™ libraries due to native wire protocol support. Such packages are readily available for installation on any host by any user – including the host that ASE is installed on.

8.3 Login Triggers

In general, login triggers were implemented as a means of providing customizable session level optimization and security controls to support `Row Level Access Controls`, etc. While these features may be needed for custom applications, they are not necessary for SAP applications.

Some customers have also used login triggers as a means of controlling which programs or hosts have access to the database instance. From a host control standpoint, network firewall capabilities provide a more secure means of isolating the database instance from external hosts. Relying on login triggers could leave the system susceptible to brute force password attacks as the login authentication is already accomplished at the point in which the login trigger is fired. Remember, gaining access to the system may not be the goal. If a perpetrator knows, for instance, that the password policy is such that three wrong passwords results in the login being locked, they may simply spam the system with wrong passwords and cause all DBA, SSO or even application accounts to be locked – resulting in an outage – which may have been their goal. So while login triggers can provide a level of comfort in restricting access, they do not provide assurance from preventing other issues.

If login triggers are used for any reason, consider the following:

- Login triggers are viewed as unnecessary and may not be supported from an SAP application viewpoint.
- Login triggers should not reference any data outside the master database. If a list of allowed applications or other access controls are created, they should be created in tables in the master database within ASE. This is critical as some recovery techniques for ASE include the ability to boot the database instance without recovering user databases or other system databases, except the master. Any reference to data outside of the master would inhibit logins as a result.
- Login triggers should not make any external OS calls, reference proxy tables/procedures via CIS or require access to any software subsystem outside the ASE instance itself. Again, any failure in those external applications would inhibit logins.

- Login triggers should contain an escape clause at the very beginning for the ASE sa login – e.g. if the login name is sa, the login trigger should simply exit.
- Changes to the login trigger should only be made when both a DBA and SSO are also connected via separate other sessions. Logic errors in a login trigger might prevent their subsequent connections, consequently, they should already be connected in order to drop the faulty login trigger.

8.4 Auditing

In general, auditing the SAPSR3 and SAPSR3DB SAP application logins is unnecessary and will only result in an overwhelming volume of audit information. However, auditing DBA and SSO activities generally is required for traceability of DBA actions for SOX/PCI/HIPPA compliance. Overall, consider the following recommendations with respect to auditing and audit trail management:

- Use multiple audit tables and set up an audit threshold (see page 370 in the *SAP ASE Security Administration Guide*) to truncate older tables prior to reuse as well as copying recently filled tables to separate database for preservation.
- Configure server auditing appropriately – increase the audit queue size to minimally be ten times the *number of user connections* configured. Disable suspend auditing when full to prevent a system outage due to problems with auditing.
- Make sure that normal transaction log thresholds are implemented for sybsecurity or truncate log on checkpoint is enabled to prevent transaction log filling from suspending auditing or access.
- Audit login failures using sp_audit login, all, fail instead of auditing logins in general. If auditing all logins, the volume could be overwhelming.
- Audit the all for DBAs and SSOs by login. While auditing system roles such as sa_role is supported, some DBA actions – such as dbcc sqltext() require sybase_ts_role and might be missed. Simply add the sp_audit call to the same script used to create the DBA logins – should be minimal overhead from an administration viewpoint.
- Grant audit reviewers permission on the audit tables using granular permissions instead of granting sso_role. If reviewing requirements allows latency (vs. alerts), then simply add them as a user to the audit trail database that the audit threshold copies data to.
- Audit DDL activity in the SID database via auditing both the CREATE and ALTER commands in that database. For example: sp_audit create, 'all', '<SID>', 'on' – note that such altering should only be enabled after migration completes to avoid overwhelming the audit log and reviewers with the auditing of the full SAP database schema.
- Consider auditing cmdtext for automated job logins. However, you may not need to audit monitoring cmdtext due to the volume – just make sure that any tools outside DBA Cockpit that do performance monitoring only have mon_role and are limited in what actions they can perform.

9 Additional Sources

The following SAP Notes may provide some additional tuning recommendations as well as configuration steps:

SAP Note:	Title:
2087322	SYB: Where to find information about SAP Applications on SAP ASE
1585981	SYB: Ensuring recoverability for SAP ASE
1588316	SYB: Configure automatic database and log backups
1724091	SYB: Performance of VBDATA and other VB/RFC tables
2049506	SYB: Tables with special optimizer treatment
1775764	SYB: Increased read performance for small lookup tables
1702338	SYB: Database hints in Open SQL for SAP ASE
1777966	SYB: Check for incomplete index statistics
2081796	SYB: ASE optimization goals supported for SAP NetWeaver-based applications
2106688	SYB: Enabling Granular Permissions in SAP ASE
1954245	SYB: Performance degradation due to high CPU usage in ASE
1668882	Note Assistant: Important notes for SAP_BASIS 730,731,740
954268	Optimization of export: Unsorted unloading
1630356	SGEN terminates after 500 generation errors
1633118	SGEN deletes GENSETM when restarting

10 Appendix

10.1 Dynamic Cache Sizing

```
use master
go
set nocount on
go
declare @mode                 varchar(30)
-- choose one of 'import' or 'runtime'
set @mode = 'runtime'
declare @phys_memory          int,
        @logical_memory       int,
        @overhead             float,
        @memory_available     int,
        @cache_increase       int,
        @default_cache_size   int,
        @new_cache_size       int,
        @new_cache_size_C     varchar(30),
        @max_engines          int,
        @global_partn         int,
        @partn_clause         varchar(30),
        @pool_size_128K       int,
        @wash_size_16K        int,
        @wash_size_128K       int,
        @cur_pool_size_16K    int,
        @cur_pool_size_128K   int,
        @cur_wash_size_16K    int,
        @cur_wash_size_128K   int,
        @param_c              varchar(30)
if @mode <> 'runtime' and @mode <> 'import'
begin
  print 'invalid mode'
  select syb_quit()
end
select @phys_memory=(scc.value*2/1024)
  from sysconfigures sc, syscurconfigs scc
   where sc.name = 'total physical memory'
    and sc.config = scc.config
select @logical_memory=(scc.value*2/1024)
  from sysconfigures sc, syscurconfigs scc
   where sc.name = 'total logical memory'
    and sc.config = scc.config
select @memory_available=@phys_memory-@logical_memory,
    @overhead=round((@phys_memory-@logical_memory)*0.062,0)
select @cache_increase=(@memory_available-@overhead)*0.9
if @cache_increase < 256
  select @cache_increase=0
-- get the size of the default data cache
select @default_cache_size=cu.value/1024
  from master.dbo.sysconfigures co, master.dbo.syscurconfigs cu
  where parent = 19
    and co.config = cu.config and co.config = 19
    and co.name = cu.comment
    and name like '%default data cache%'
select @new_cache_size=@default_cache_size+@cache_increase
select @new_cache_size_C=convert(varchar(30),@new_cache_size)+'M'
select @logical_memory as logical_memory,
       @phys_memory as physical_memory,
       @default_cache_size as default_cache_size,
       @cache_increase as cache_increase,
       @new_cache_size as new_cache_size,
       @new_cache_size_C as new_cache_size_C
if @cache_increase > 0
  exec sp_cacheconfig 'default data cache', @new_cache_size_C
select @max_engines=scc.value
  from sysconfigures sc, syscurconfigs scc
  where sc.name='max online engines'
    and sc.config = scc.config
select @global_partn=power(2,round(log(@max_engines)/log(2),0))
select @partn_clause='cache_partition='+convert(varchar(10),@global_partn)
select @max_engines as max_engines,
       @global_partn as global_partn,
       @partn_clause as partn_clause
-- Partition default data cache to reduce spinlock contention. The general rule
is:
-- cache partitions >= number of threads in syb_default_pool, but being within
-- 8, 16, 32 or 64):
exec sp_cacheconfig 'default data cache', @partn_clause
-- get the size of the new default data cache
select @default_cache_size=cu.value/1024
  from master.dbo.sysconfigures co, master.dbo.syscurconfigs cu
  where parent = 19
    and co.config = cu.config and co.config = 19
    and co.name = cu.comment
    and name like '%default data cache%'
-- During import, we use 50% of default data cache for large IO
-- During runtime, we use 20% of default data cache for large IO
if @mode = 'import'
  select @pool_size_128K=cast((@default_cache_size*0.5) as integer)
else
  select @pool_size_128K=cast((@default_cache_size*0.2) as integer)
select @param_c=convert(varchar(10),@pool_size_128K)+'M'
select @default_cache_size as default_cache_size,
       @pool_size_128K as pool_size_128K, @param_c as param_c
exec sp_poolconfig 'default data cache', @param_c, '128K'
select @cur_pool_size_16K=value/1024, @cur_wash_size_16K=memory_used
  from master.dbo.syscurconfigs
 where config = 20 + (log10(16*1024/@@pagesize)/log10(2))
   and comment = 'default data cache'
select @cur_pool_size_128K=value/1024, @cur_wash_size_128K=memory_used
  from master.dbo.syscurconfigs
 where config = 20 + (log10(128*1024/@@pagesize)/log10(2))
   and comment = 'default data cache'
-- Each cache pool has a wash area, which determines the point in the LRU chain
-- when ASE will consider a dirty page for background write to disk.
-- During import, we set the wash markers at 50% of each pool
-- During runtime, we set the wash markers at around 5-10% of each pool

if @mode = 'import'
begin
  select @wash_size_16K=cast((@cur_pool_size_16K*0.5) as integer)
  select @wash_size_128K=cast((@cur_pool_size_128K*0.5) as integer)
end
else
begin
  select @wash_size_16K=cast((@cur_pool_size_16K*0.1) as integer)
  select @wash_size_128K=cast((@cur_pool_size_128K*0.1) as integer)
end
select @param_c='wash='+convert(varchar(10),@wash_size_16K)+'M'
select @default_cache_size as default_cache_size, @cur_pool_size_16K as
pool_size_16K, @wash_size_16K as wash_size_16K
```

```
exec sp_poolconfig 'default data cache', '16K', @param_c
select @param_c='wash='+convert(varchar(10),@wash_size_128K)+'M'
select @default_cache_size as default_cache_size, @cur_pool_size_128K as
pool_size_128K, @wash_size_128K as wash_size_128K
exec sp_poolconfig 'default data cache', '128K', @param_c
-- During import, we set an APF ratio of 50% for any cache in the system
-- During runtime, we set an APF ratio of 10% for any cache in the system
if @mode = 'import'
  exec sp_configure 'global async prefetch limit', 50
else
  exec sp_configure 'global async prefetch limit', 10
go
```

10.2 Check Completeness of Table Statistics

The following SQL script identifies any indexed table column that does not have optimizer statistics:

```
use <SID>
go
SELECT tabowner, tabname, keycnt, row_count(db_id(),t3.objid) as rowcnt
  FROM (
    SELECT cast(tabowner as varchar(10)) as tabowner,
           cast(tabname as varchar(30)) as tabname,
           count(*) AS keycnt,
           objid
     FROM (
       SELECT DISTINCT t1.tabowner, t1.tabname, t1.objid, t1.indexcol
         FROM ( SELECT su.name AS tabowner, so.name AS tabname,
                       so.id AS objid, st.partitionid,
                       index_col(su.name || '.' || so.name, si.indid, sc.colid)
                       AS indexcol
                  FROM sysobjects so, sysindexes si, syscolumns sc,
                       sysusers su, systabstats st
                 WHERE so.uid = su.uid AND so.id = si.id
                   AND so.id = sc.id AND so.type = 'U'
                   AND st.id = so.id AND st.indid = 0 AND st.rowcnt > 0
                   AND so.sysstat3 & 128 = 0
                   AND si.indid > 0 AND si.indid < 255
                   AND sc.colid <= si.keycnt
               ) t1
         WHERE indexcol IS NOT NULL
       ) t2
WHERE NOT EXISTS ( SELECT 1 FROM sysstatistics ss, syscolumns sco
                    WHERE ss.id = t2.objid AND ss.id = sco.id
                      AND sco.name = t2.indexcol
                      AND convert(smallint,ss.colidarray) = sco.colid
                      AND ss.formatid = 100 AND ss.c4 IS NOT NULL
                  )
  AND row_count(db_id(),t2.objid) > 100
GROUP BY tabowner, tabname, objid ) t3
go
```

If this SQL script returns any row, you should create index statistics for each table listed manually:

```
tabowner    tabname                      keycnt     rowcnt
---------- ------------------------------ ----------- --------------------
SAPSR3      AGR_FLAGS                     1         3000
SAPSR3      /BIC/B0000176000              1         12177665
SAPSR3      /BIC/FMADHVI01                3         10100
update index statistics SAPSR3.[AGR_FLAGS] with hashing
```

```
go
update index statistics SAPSR3.[/BIC/FMADHVI01] with hashing
go
-- use consumers due to high row count:
update index statistics SAPSR3.[/BIC/B0000176000] with consumes = 3, hashing
go
```

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: https://help.sap.com/viewer/disclaimer).

go.sap.com/registration/contact.html